NPS72-86-004CR

NAVAL POSTGRADUATE SCHOOL
Monterey, California

CONTRACTOR REPORT

IMPULSIVE LOADING FROM A BARE

EXPLOSIVE CHARGE IN SPACE

by

Joseph Falcovitz

December 1986

Approved for public release; distribution unlimited.

Prepared for: Strategic Defense Initiative Office
The Pentagon
Washington, DC 20301-7100

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A180 350

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS: NONE
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for Public Release; Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): NPS72-86-004CR	5. MONITORING ORGANIZATION REPORT NUMBER(S): NPS72-86-004CR
6a. NAME OF PERFORMING ORGANIZATION: JOSEPH FALCOVITZ; 6b. OFFICE SYMBOL (If applicable): 72	7a. NAME OF MONITORING ORGANIZATION: NAVAL POSTGRADUATE SCHOOL, CODE 72
6c. ADDRESS (City, State, and ZIP Code): Research Contractor, Naval Postgraduate School, Code 72, Monterey, CA 93943-5100	7b. ADDRESS (City, State, and ZIP Code): Space Systems Academic Group, Monterey, CA 93943-5100
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Strategic Defense Initiative Office; 8b. OFFICE SYMBOL (If applicable): SDIO/DEO	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: MIPR-DGAA60045
8c. ADDRESS (City, State, and ZIP Code): SDIO/DEO, Washington, DC 20301-7100	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. PE63221; PROJECT NO.; TASK NO.; WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

Impulsive Loading From A Bare Explosive Charge in Space

12. PERSONAL AUTHOR(S)
JOSEPH FALCOVITZ

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Contractor Report	FROM Jan 86 TO June 86	December 1986	65

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Space Blast, Impulsive Loading, Explosive in Space

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Consider a planform target subjected to a normal impact of explosive products generated by detonating a bare charge in space. It is suggested that the loading impulse may be approximated by the total momentum of that portion of the fluid which impacts at the target. Assuming impulsive dynamic response, and assuming that the ensuing damage is proportional to the kinetic energy imparted to the structure by the blast, we get a particularly simple law: Damage $\sim \mathbf{W}^2/\mathbf{R}^4$ (**W** is charge mass, **R** is range). This model is an idealization of a solar panel (or antenna) extended in a paddle-like fashion from a relatively rigid and massive core structure. It is also shown that this law implies that no advantage can be realized by re-arranging the mass of a single bare charge in a cluster configuration of smaller sub-charges, which would be dispersed and detonated via an idealized "isotropic" scheme.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: ALLEN E. FUHS, Distinguished Professor	22b. TELEPHONE (Include Area Code): (408)646-2948; 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

U.S. Government Printing Office: 1985—606-24.

to the 2d power

Similar to

to the 4th power

ABSTRACT

This document

Considers a planform target subjected to a normal impact of explosive products generated by detonating a bare charge in space. It is suggested that the loading impulse may be approximated by the total momentum of that portion of the fluid which impacts at the target. Assuming impulsive dynamic response, and assuming that the ensuing damage is proportional to the kinetic energy imparted to the structure by the blast, we get a particularly simple law: Damage $\sim W^2/R^4$ (W is charge mass, R is range). This model is an idealization of a solar panel (or antenna) extended in a paddle-like fashion from a relatively rigid and massive core structure. It is also shown that this law implies that no advantage can be realized by re-arranging the mass of a single bare charge in a cluster configuration of smaller sub-charges, which would be dispersed and detonated via an idealized isotropic scheme.

Keywords: gas dynamics; exhaust plumes; Spacecraft.

ACKNOWLEDGEMENTS

This work is part of a study involving gas dynamics of exhaust plumes from spacecrafts. It was conducted under the cognizance of Distinguished Professor Allen E. Fuhs, who suggested extending our understanding of gasdynamics in space to the treatment of blast effects on spacecrafts. I wish to thank Professor Fuhs for his creative guidance and deeply appreciate his continuous support. The GRP code used for the blast computation is a product of mutual research conducted by Professor M. Ben-Artzi and myself. The fruitful collaboration of Professor Ben-Artzi is gratefully acknowledged.

TABLE OF CONTENTS

1. INTRODUCTION 1
2. IMPACT BLAST LOADING 3
3. TARGET DYNAMIC RESPONSE 10
4. CLUSTER CONFIGURATION 13
5. DISCUSSION AND CONCLUSIONS 15
6. REFERENCES 16

APPENDIX A. The GRP Code 17
- A.1 Array Variables 18
- A.2 Major Parameters 20
- A.3 Labeled COMMON variables 21
- A.4 Description of Subroutines 24
- A.5 Listing of GRP Code 29

APPENDIX B. Code for Re-Normalizing the Air Impulse 56

7. DISTRIBUTION LIST 57

LIST OF FIGURES

Figure 2-1 Impact Blast Loading 7
Figure 2-2 Shock Reflection at Impact Phase 8
Figure 2-3 Limiting Cases of Shock Reflection
(a) Initially Reflected Shock (Impact) (b) Stationary Shock 8
Figure 2-4 Impulse of Normally Reflected Blast Wave at Sea-Level and in Space 9
Figure 3-1 Cantilever Beam with Plastic Hinge 12
Figure 3-2 ChargeMass - Range - Damage Curves for Cantilever Beam 12
Figure 4-1 Target Intercept at Closest Approach 14
Figure 4-2 Spherical Cap Surrounding the Target 14
Figure A-1 Piecewise Linear Distribution of Flow Variables in Cells 53
Figure A-2 Intersection of Right and Left Adiabats for Solving Riemann Problem 54
Figure A-3 Wave Diagram Representing Solution to Riemann Problem 55

NOMENCLATURE (consistent units in m, kg, ms system)

C	Coefficient in ChargeMass-Range-Damage relationship ($m\ kg^{-1/2}$)
D_{CJ}	Speed of propagation of detonation wave at CJ point ($m\ ms^{-1}$)
I	Impulse per unit area of target ($kg\ m^{-1}\ ms^{-1}$)
$\hat{I}$	Dimensionless impulse $\hat{I} = I(R)\ [4\pi R_0{}^2/W(2Q_0)^{1/2}]$
h	Beam thickness (m)
L	Length of cantilever beam (m)
m	Lagrange mass coordinate (kg)
M_p	Moment per unit length of plastic hinge ($MPa\ m^2$)
N	Number of sub-charges in a cluster configuration
P	Pressure (MPa)
P_s	Surface pressure (MPa)
Q_0	Explosive energy per unit mass ($MJ\ kg^{-1}$)
R_0	Radius of spherical charge (m)
R	Range from center of charge (m)
S	Speed of propagation of shock wave ($m\ ms^{-1}$)
t	Time (ms)
U	Flow velocity ($m\ ms^{-1}$)
V	Velocity imparted to target by loading impulse ($m\ ms^{-1}$)
W	Charge mass (kg)
Y	Plastic yield stress (MPa)
Z	Total momentum of an explosive charge ($kg\ m\ ms^{-1}$)
α	Coefficient for dynamic pressure recovery
γ	Specific-heat ratio
γ_{CJ}	Specific-heat ratio of explosive products at CJ point
θ	Plastic rotation angle of cantilever beam
κ	Impact approximation impulse coefficient (presently $\kappa = 1$)
μ	Beam mass per unit area ($kg\ m^{-2}$)
ρ	Fluid density ($kg\ m^{-3}$)
ρ_p	Beam density ($kg\ m^{-3}$)
φ	Mid-area angle of sub-charge spherical cap

EMPTY PAGE

1. INTRODUCTION

The advent of space-based weapon systems in our times has raised the prospects of future "Star Wars" conflicts, rendering the potential use of explosive devices against space targets a present day engineering reality. The warhead of choice in space seems to be of the fragmentation type, for obvious reasons. The effectiveness of fragments is unhampered by the space environment (lack of air may even be helpful). By contrast, bare charges in space are considerably less efficient than in air. One may wonder why this is so since in air, as in space, the same amount of chemical energy is released through the detonation process. The explanation is that the difference is in the much larger mass involved in the air blast, relative to the bare charge mass.

For a more comprehensive explanation, we take a close look at the process by which an explosive-driven air blast wave is generated. The explosive products effectively constitute a rapidly expanding spherical piston (typical initial speed around 6 km/sec), which drives an intense shock wave into the surrounding air. At a typical range of $100\mathbf{R_0}$ (and with air density equal to about 1/1000 of charge density), the mass of air entrained by the shock is about 1000 times the charge mass. Thus, the highly concentrated initial explosive energy, has spread over a much larger mass than that of the charge, via the mechanism of wave propagation in compressible media, resulting in an increased momentum. For a comprehensive treatment of blast waves in air the reader is referred to Baker[1].

It is also worthwhile noting that explosive products in space typically attain hypersonic speed prior to impacting at the target. The flow velocity in an air blast is typically subsonic or somewhat supersonic. It is thus expected that the actual gasdynamic interaction between the blast flow and a stationary target, will be fundamentally different in these two cases.

We contend that blast effects in space may still be of practical interest for reasons such as the following :

(i) Notwithstanding the poor efficiency of a bare charge, its use should not be ruled out altogether. Fragments would contribute to existing - and potentially hazardous - population of space debris, underlining the obvious fact that there is no absolutely safe standoff distance from an isotropic fragmentation warhead. A clean bare charge may thus be a reasonable alternative.

(ii) Even a fragmentation warhead has some residual blast capacity, which has to be considered either as a factor in enhancing target damage, or as a threat to be reckoned with in determining a safe standoff distance.

The key idea of the present model is a combination of the assumption that target dynamic response is related primarily to total blast impulse, and the physically plausible notion that this impulse is equal to the total momentum of that portion of the expanding explosive products which impacts at the target. The sense in which this simple notion constitutes an approximation to a proper gasdynamic analysis of the interaction between the fluid and the target, is clarified in Ch. 2. In that chapter we also present an illuminating comparison between impulsive blast loading in air and in space.

In order to demonstrate the ChargeMass-Range-Damage relationship implied by our impact blast approximation, we chose a simple target model: A cantilever beam with a rigid-perfectly plastic stress-strain relationship. It represents an extended structural element such as a solar panel or an antenna. We make use of studies conducted by Mentel [2] and by Bodner and Symonds [3], which showed that by and large, the effect of accelerating the beam impulsively was to cause a rotation about a plastic hinge at the point of support. The final angle of rotation is generally proportional to the initial kinetic energy, so that equating damage with that angle, results in damage being proportional to the square of the impulse imparted to the target by the blast loading. A presentation of this dynamic response model, including a sample case, is given in Ch. 3.

Our ChargeMass-Range-Damage relationship may imply some far-reaching conclusions when applied to the analysis of a more general configuration than the single-charge/single-target case. In Ch. 4 we present a simple analysis of a sub-munition configuration of N bare charges, concluding that it seems to have no advantage in efficiency, relative to a single charge of equal mass. Sections 5 and 6 contain conclusions and references, correspondingly.

We conclude the introduction by listing the main assumptions made in the present study:

(a) Blast loading and target response are uncoupled. This is true since typically the target mass is much larger than the mass of that portion of the explosive products which impacts on it.

(b) Dynamic target response is independent of specific loading time history. It depends solely on total (time-integrated) impulse.

(c) The target is a panel extended as a relatively supple cantilever. It is supported by a relatively rigid and massive core structure.

(d) The charge is a sphere detonated at its center. The expansion is spherically symmetric.

(e) Target surface is normal to local flow vector.

(f) Target orbital velocity relative to the center of the charge is negligible, compared with the velocity of the expanding products.

2. IMPACT BLAST LOADING

Consider the expanding explosive products impacting at a target as shown in Fig. 2-1. By regarding the fluid as an ensemble of non-interacting particles moving at velocity $U(R,t)$, and by assuming a no-rebound normal impact at the surface, the pressure time history is given by :

$$P_s(t) = \rho(R,t)[U(R,t)]^2 \tag{2-1}$$

How is this simple impact mechanism related to the actual gasdynamic interaction between the expanding explosive products and the target? When a target is located at a range of at least several charge radii, two features in the free stream of the oncoming fluid are significant : The flow is highly hypersonic (Mach number 20 or higher), and the static pressure is very small, which means that $P + \rho U^2 \approx \rho U^2$. These facts were born out by a numerical computation which we performed for a typical high explosive characterized by the following parameters :

$$\begin{aligned}
\rho_0 &= 1800 \quad (\mathrm{kg\ m^{-3}}) \\
\gamma_{CJ} &= 3 \\
D_{CJ} &= 8 \quad (\mathrm{m\ ms^{-1}}) \\
Q_0 &= D_{CJ}^2/[2(\gamma_{CJ}^2 - 1)] = 4 \quad (\mathrm{MJ\ kg^{-1}})
\end{aligned} \tag{2-2}$$

Where Q_0 was determined by assuming that the detonation corresponded to the CJ point on the explosive Hugoniot curve, and that the detonation products were an ideal gas with a specific-heat ratio γ_{CJ}. The spherically expanding flow was computed by integrating the Euler equations for isentropic flow via a high-resolution conservative finite-difference scheme [4-6]. The initial conditions were the self-similar flow field of a just-detonated spherical charge given by Taylor [7]. The code GRP with which the computation was performed is described and listed in Appendix A.

Consider the flow at a stationary target, which begins at the moment of arrival of the expanding explosive products (Fig. 2-2). A qualitative description of the ensuing flow pattern is made by observing its evolution in time. Immediately following the initial (normal) impact, the fluid is stopped at the target by a backward-propagating shock wave reflected from the surface. Since the target is of

finite extent, the fluid between the shock and the surface is accelerated laterally, and streamlines that tend to curve around the target are being formed. If the oncoming flow were stationary, the flow field would evolve toward the familiar configuration of a detached bow-shock positioned at a relatively narrow standoff distance from the surface.

Let us find the post-shock pressure in these two limiting phases. In the initial phase, the fluid is stopped at the target by a reflected shock (Fig. 2-3a), and in the pseudo-stationary phase (Fig. 2-3b), the shock is stationary. In either case we find the post-shock pressure to be given by a pressure-recovery expression of the form :

$$P_2 = \alpha \rho U^2 \tag{2-3}$$

Where α is a constant related to the appropriate γ (assuming the expanded explosive products are an ideal gas). The governing equations in the reflected shock case are :

$$\begin{aligned} &\rho(U+S) = \rho_2 S \\ &\rho(U+S)^2 = P_2 \\ &\rho(\gamma+1)/(\gamma-1) = \rho_2 \qquad \text{(strong shock)} \end{aligned} \tag{2-4}$$

Where the unknowns are ρ_2 , P_2 , S .

The equations for the stationary shock case are :

$$\begin{aligned} &\rho U = \rho_2 U_2 \\ &\rho U^2 = P_2 + \rho_2 U_2^2 \\ &\rho(\gamma+1)/(\gamma-1) = \rho_2 \qquad \text{(strong shock)} \end{aligned} \tag{2-5}$$

Where the unknowns are ρ_2 , U_2 , P_2 . Thus, solving for α in the two cases represented by equations (2-4) and (2-5), we get :

$$\text{Reflected shock} \quad \alpha = [(\gamma+1)/2]^2$$

$$\text{Stationary shock} \quad \alpha = 2/(\gamma+1) \tag{2-6}$$

In either case, since the gas is not dense, the effective range of γ is somewhere between 1.0 and 1.4, so that setting $\alpha = 1$ is an approximation commensurate with the overall crudeness of the present impact blast model. Since the flow in the layer between the shock and the target is low subsonic (at least it is so away from target edges), the post-shock pressure is a reasonable substitute for the surface pressure. Also, $\alpha = 1$ is an appropriate approximation where the flow is so rarefied that it is collisionless. In this limit, $\alpha = 1$ corresponds to full thermal accommodation of re-emitted molecules from a presumably cold surface.

The foregoing analysis constitutes a justification of the impact approximation to the surface pressure (2-1). Now we turn to the task of evaluating the impulse which is defined as the time-integrated surface pressure. Using the impact approximation (2-1), the impulse is given by :

$$I(R) = \int_0^\infty P_s(t)dt = \int_0^\infty \rho(R,t)[U(R,t)]^2 dt \tag{2-7}$$

Let us introduce a Lagrange mass coordinate m which enables a transformation from the Euler system (R,t) to the Lagrange system (m,t). The differential relation associated with this transformation at constant R is :

$$dm = 4\pi R^2 \rho(R,t)U(R,t)dt \tag{2-8}$$

Since it is assumed that the fluid is not accelerated at any (R,t) in the range of interest for blast loading, the velocity $U(R,t)$ can be regarded as function *solely of the mass coordinate*, so that $U(R,t) = U(m)$. Using (2-8) we are then able to cast the impact blast expression (2-7) in the following simple and physically appealing form :

$$I(R) = Z/4\pi R^2$$

$$Z = \int_0^W U(m)dm \tag{2-9}$$

The upper limit W in (2-9), which is consistent with the upper limit ∞ in (2-7), implies that the total impulse is somewhat overestimated, since it contains contributions from the innermost layers of the explosive products that will arrive at the target as $t \rightarrow \infty$.

The total momentum Z is thus a constant which can be evaluated for any specific explosive charge by numerical integration. We performed this computation with the code GRP described in Appendix A. In doing so for the typical explosive (2-2), we found out that the impulse (2-9) was a reasonable approximation at ranges as low as $R = 3R_0$. Furthermore, it was found that Z could be approximated by the maximum attainable momentum for the given charge mass and energy $W(2Q_0)^{1/2}$, to within about 6%. Apparently, the total momentum is not overly sensitive to the exact velocity distribution function $U(m)$, so that assuming a value of Z appropriate to the uniform distribution $U(m) = (2Q_0)^{1/2}$ is a reasonable approximation. Thus we finally arrive at the following closed-form approximation for the blast impulse :

$$I(R) = \kappa W(2Q_0)^{1/2}/4\pi R^2 \qquad \kappa = 1 \tag{2-10}$$

Where the coefficient κ is retained in order to suggest that its value be determined more accurately from detailed experimental or computational data, in the event that such data become available. At present our best estimate is $\kappa = 1$.

There is one comparison, however, which can readily be made with available data. We refer to impulsive blast loading in air, such as given by Baker (Ref. 1, Fig. 6.3 in the supplement). The comparison is conveniently made with a non-dimensional form of (2-10), which is rewritten as :

$$\bar{I} = I(R)\,[4\pi R_0^2/W(2Q_0)^{1/2}] = (R/R_0)^2 \tag{2-11}$$

The air blast data has to be converted to the same normalization scheme as in Eq. (2-11), before the comparison can be made. Considering the definition of $\bar{I}$ in (2-11) above, and the definition of scaled range and air blast impulse (Table 6.2 of Ref. 1), this conversion is done by multiplying the scaled air impulse and range by the following coefficients (sea-level air is assumed) :

Impulse Multiplier $\qquad \beta = 3(2\gamma)^{-1/2}(4\pi/3)^{1/3}(P_a/\rho_a Q_0)^{1/6}(\rho_a/\rho_0)^{1/2} = .01204$

Range Multiplier $\qquad \delta = (4\pi/3)^{1/3}(\rho_0 Q_0/P_a)^{1/3} = 67.06$ $\qquad$ (2-12)

$\rho_a = 1.3$ (kg m^{-3}) $\qquad P_a = 0.1$ (MPa) $\qquad \gamma = 1.4$

The air blast conversion was done by a small code which is given in Appendix B. The air and space blast impulses are shown in Fig. 2-4. We note that at ranges larger than about 10 charge radii, the air blast impulse is higher than the space impulse, and the gap widens as the range increases. This observation is consistent with the qualitative explanation given in the introduction, which attributed this effect to the increase in the entrained air mass at higher range. At ranges lower than 10 charge radii, the air mass is relatively insignificant, so that one may expect the blast impulses in air and in space to be comparable. Indeed, the inverse-square variation of impulse with range is apparent for the air blast at low range. In absolute values, however, the low-range space impulse is higher by a factor of about 1.7. This might be interpreted as indicating that choosing $\kappa = 1/1.7$ would be the appropriate "calibration". However, we do not propose to do so, since we are not able to trace the various factors affecting the low-range impulse as given by Baker [1]; they may somehow depend on the presence of air, as well as on other parameters such as target size and equation of state of the explosion products.

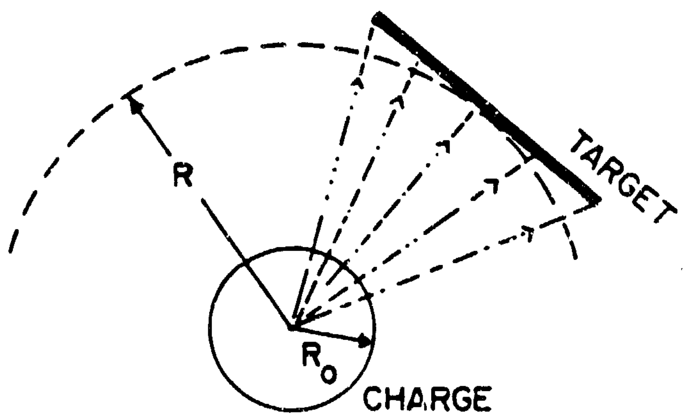

Figure 2-1. Impact Blast Loading

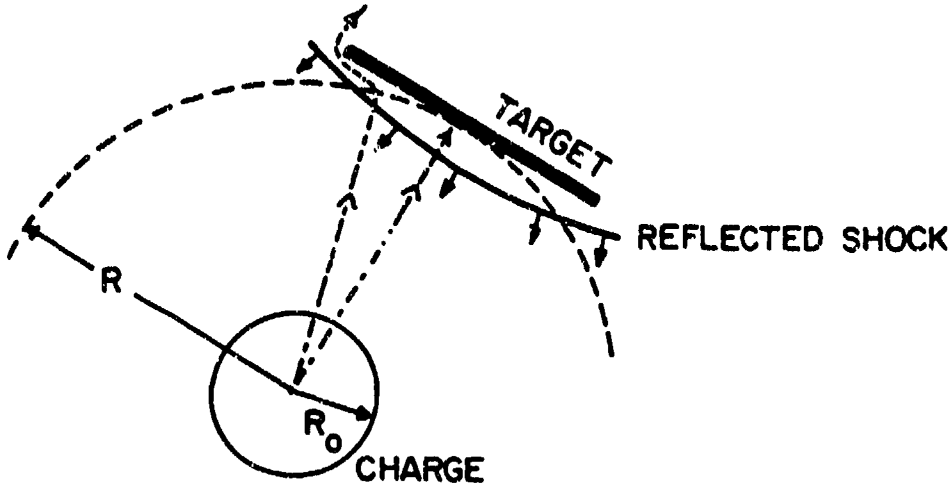

Figure 2-2. Shock Reflection at Impact Phase

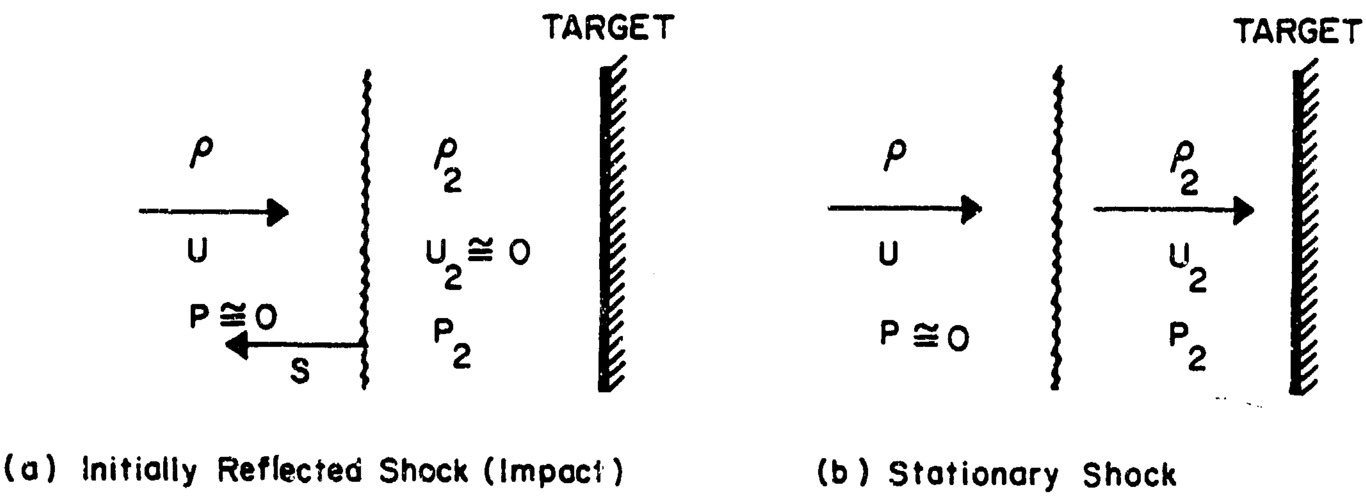

Figure 2-3. Limiting Cases of Shock Reflection

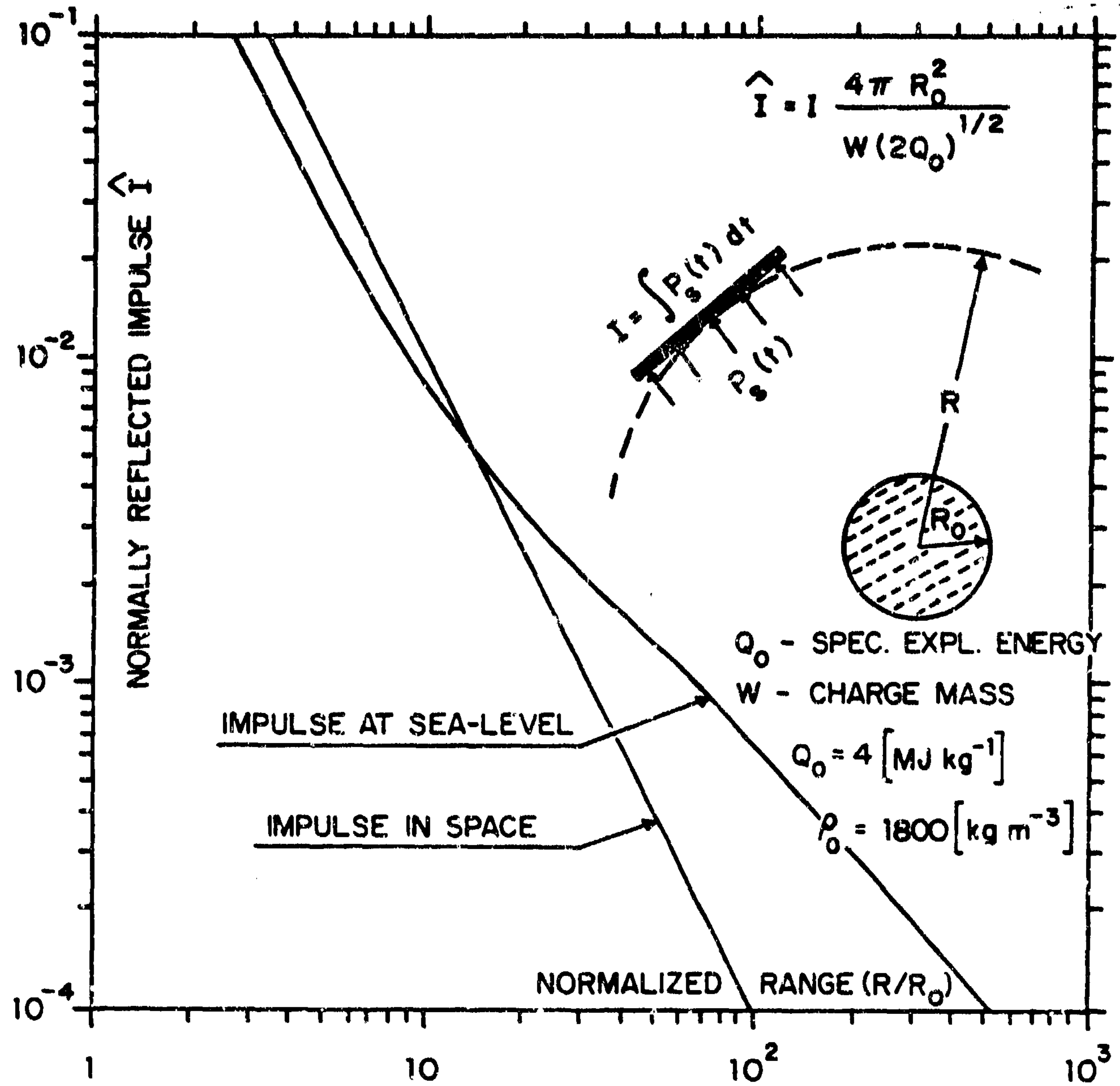

Figure 2-4. Impulse of Normally Reflected Blast Wave at Sea-Level and in Space

3. TARGET DYNAMIC RESPONSE

For the sake of constructing representative ChargeMass-Range-Damage relations from our impact approximation to the blast impulse (2-10), we suggest a simple idealized structure as target model. It is a cantilever beam made of a metal characterized by a rigid-perfectly plastic stress-strain relation.

This model is supposed to represent an extended spacecraft component such as a solar panel or an antenna. The core structure is assumed to be much more massive and rigid than the extended structural element, so that the cantilever can be idealized as being rigidly supported. The sole dynamic and structural parameters are hence those of the cantilever.

For this purpose we make use of an experimental and theoretical investigation of uniform cantilever beams subjected to impulsive loading that was conducted by Mentel [2]. Aluminum alloy beams were held in a massive support that was gliding along a rail at speed V, until it was abruptly stopped by a very massive anvil. After the system came to rest, the beams were observed to have rotated through an angle θ about the point of support, with little deformation elsewhere (Fig. 3-1).

The theoretical model suggested by Mentel [2] for predicting $\theta(V)$, can be described as comprising two stages. Immediately following the impact, the beam commences rotating rigidly about the support point, with an angular momentum equal to the pre-collision moment of momentum about that point. This application of the principle of conservation of moment of momentum entails an abrupt re-distribution of velocity in the beam, with velocity being proportional to distance from support, and the tip moving at $1.5\ V$. The angle θ is subsequently determined from the requirement that the rotational kinetic energy be dissipated as plastic hinge work $M_p\theta$. The resulting $\theta(V)$ expression is :

$$\theta = (3/8)\mu L V^2/M_p \tag{3-1}$$

We now make one more step in formulating the model, in that we postulate that ***the angle*** θ ***is a measure of damage***. Using the following expressions for M_p, μ and V :

$$M_p = (1/4)Yh^2$$

$$\mu = \rho_p h \tag{3-2}$$

$$V = I(R)/\mu$$

We get from (2-10) and (3-1) the following ChargeMass-Range-Damage (W-R-θ) relationship :

$$R = CW^{1/2}$$

$$C = [(3/16\pi^2\theta)\,(LQ_0/\rho_p Yh^3)]^{1/4} \qquad (3\text{-}3)$$

We note that the effective range for a specified target and "damage level" θ, is proportional to the square root of the charge mass W.

Using the data for the typical explosive (2-2), and the following data for a specific aluminum beam, we get for this sample case :

$$h = 0.002 \quad (\mathrm{m})$$

$$L = 1.0 \quad (\mathrm{m})$$

$$\rho_p = 2700 \quad (\mathrm{kg\ m^{-3}}) \qquad (3\text{-}4)$$

$$Y = 300 \quad (\mathrm{MPa})$$

$$C = 1.85\,\theta^{-1/4} \quad (\mathrm{m\ kg^{-1/2}})$$

The ChargeMass-Range-Damage relationship corresponding to this sample case is depicted in Fig. 3-2.

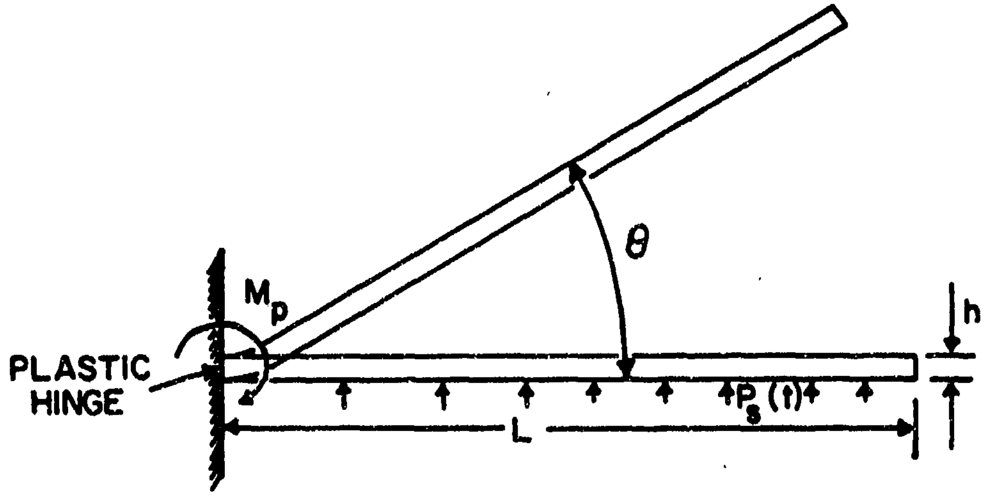

Figure 3-1. Cantilever Beam with Plastic Hinge

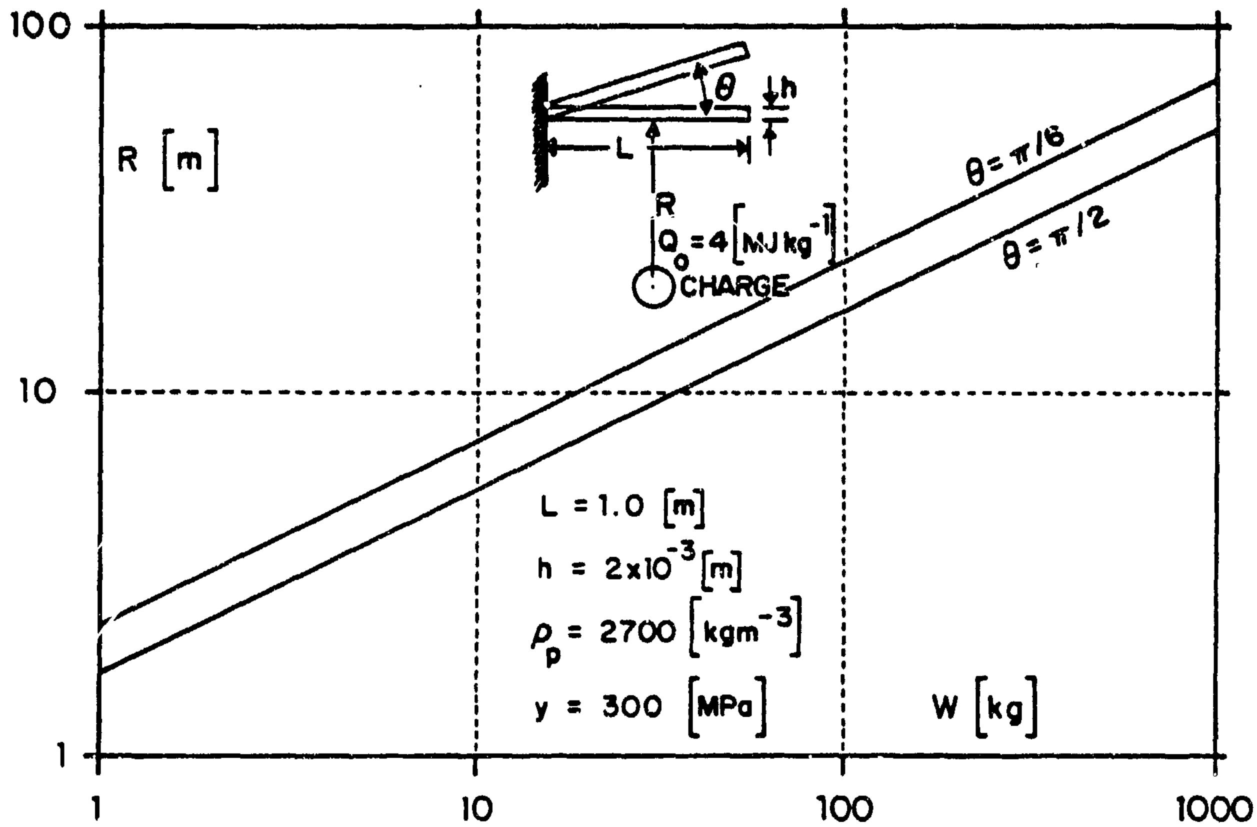

Figure 3-2. ChargeMass - Range - Damage Curves for Cantilever Beam

4. CLUSTER CONFIGURATION

In a cluster configuration, the gain in damage is presumably a result of a favorable design tradeoff between reduced charge mass and reduced range. Can such a gain be achieved for a space system, assuming the ChargeMass-Range-Damage law (3-3) to hold? It can be shown that by adopting some simple strategy of sub-munition dispersion and initiation, equation (3-3) implies no gain in target damage.

Let us assume for the sake of a reasonably simple analysis, that dispersion and initiation of sub-charges would take place according to the following scheme :

(a) The N sub-charges appear to fan out from a common virtual center, moving at equal speeds. At subsequent times, their centers are uniformly distributed over an expanding spherical envelop.

(b) The target moves at a constant velocity relative to the virtual center. Its point of closest approach to that center is at range **R** .

(c) The timing for dispersion is chosen so that the target intersects (tangentially) with the spherical envelop at the point of closest approach (Fig. 4-1). This is also the point at which the blast from a single-charge configuration detonated at the virtual center, would have impacted at the target.

(d) All sub-munitions are detonated at this "moment of closest approach".

(e) It is assumed that each spherical cap of area $4\pi R^2/N$ will contain one, and only one, sub-charge. The probability of the charge location on that cap is assumed to be uniformly distributed. The expected location on the cap is hence that latitude line φ which divides the cap into two parts of equal area (Fig. 4-2).

(f) It is assumed that the target is subjected to the blast of a single sub-charge, which is located on the mid-area latitude φ of the spherical cap that surrounds the target (Fig. 4-2).

Since the area of the spherical cap subtended by φ is $4\pi R^2/(2N)$, the angle φ is given by :

$$\sin(\varphi/2) = (2N)^{-1/2} \tag{4-1}$$

We seek a comparison between the deflection θ for a single charge (W,R) , and the deflection θ_N in the sub-munition case ($W_N = W/N$, $R_N = 2R\sin(\varphi/2)$). From the ChargeMass-Range-Damage law (3-3), using also Eq. (4-1), we get :

$$(\theta_N/\theta) = (W_N/W)^2 (R/R_N)^4 = 1/4 \qquad (4\text{-}2)$$

Consequently, there is no potential gain in a tradeoff between charge mass and range, for a cluster configuration with the aforementioned dispersion scheme. The factor 1/4 , along with the mass overhead inherent in constructing a multi-charge configuration, indicate that in causing blast damage, a single charge is more effective than an equal-mass isotropically dispersed cluster.

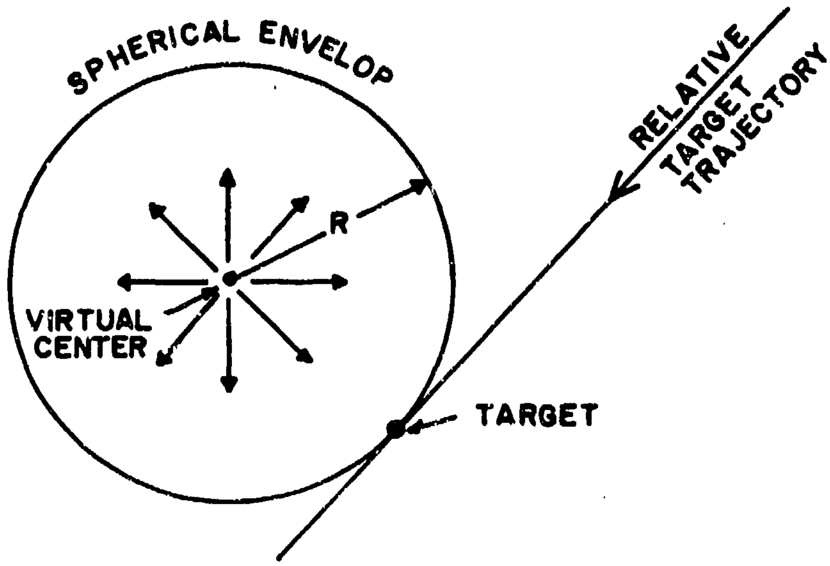

Figure 4-1. Target Intercept at Closest Approach

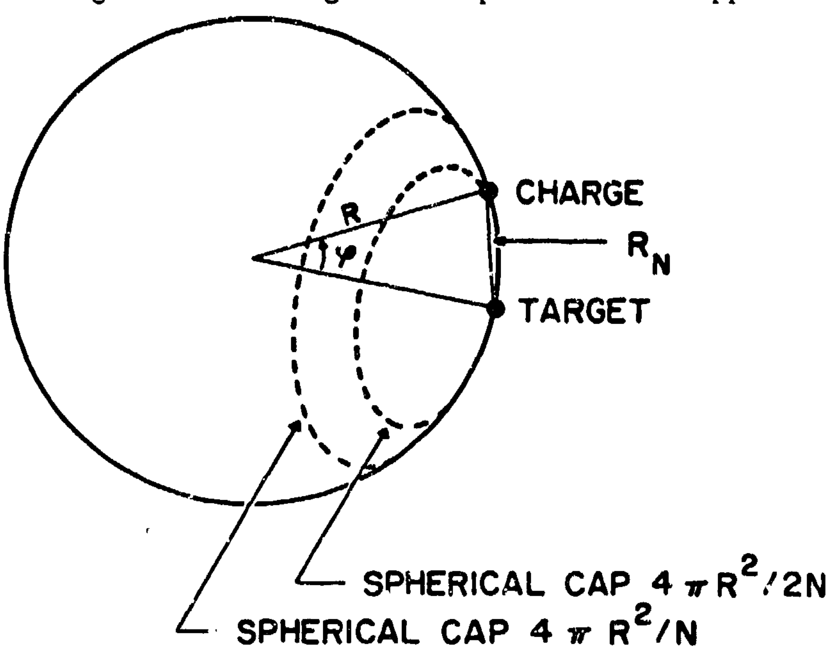

Figure 4-2. Spherical Cap Surrounding the Target

5. DISCUSSION AND CONCLUSIONS

Our analysis pertains to a bare explosive charge initiated at a point of closest approach to the target. We have shown that the loading impulse on a planform target is given by the impact approximation (2-7), which states that the impulse is proportional to the charge mass and inversely proportional to the range squared. The impulse in space has been compared with impulse in air at sea-level. It was found that the two are quite comparable at close range (10 charge radii or less), exhibiting identical variation with range. At far ranges, the impulse in air is the higher one. This is consistent with the notion that spreading the explosive energy over larger air mass results in larger momentum (and hence reflected impulse). We then proceeded to develop the ChargeMass-Range-Damage law (3-3) for an impulse-responsive target, which states that blast damage is proportional to the square of the charge mass and inversely proportional to the fourth power of the range. These results were obtained by introducing extensive simplifications in the analysis of gasdynamic interaction, and in the analysis of dynamic target response. We have further shown that this damage law also implies that no gain can be achieved by an idealized cluster configuration of bare sub-charges, relative to a single charge of equal total mass.

It is worthwhile noting that all assumptions introduced in the course of formulating the impact blast approximation and the structural dynamic response to impulsive loading, imply that target damage is overestimated. The only exception is the approximation in setting $\alpha = 1$, which can be readily rectified by assigning to α the reflected shock value given in (2-6). Furthermore, we assumed that the pressure at the midpoint of the target, is the pressure everywhere on the target. Due to flow around the edges, the average pressure is lower than the midpoint pressure. Also, targets are not everywhere normal to the flow (and charge/target attitude is not a design parameter). Oblique impact obviously entails reduced target loading. In the area of structural dynamic response, a time-distributed loading function generally delivers less kinetic energy to the structure than an impulsive loading of equal total impulse, resulting in reduced deformation (damage). Thus, while the present model may be regarded as an over estimate when applied to a sure-fail analysis, it is particularly suitable in determining a sure-safe range.

6. REFERENCES

[1] Baker, W. E., Explosions in Air , University of Texas Press, Austin and London, 1973.

[2] Mentel, T. J., "The Plastic Deformation Due to Impact of a Cantilever Beam with an Attached Tip Mass", Journal of Applied Mechanics, pp. 515-524, 1958.

[3] Bodner, S. R. and Symonds, P. S., "Experimental and Theoretical Investigation of the Plastic Deformation of Cantilever Beams Subjected to Impulsive Loading", Journal of Applied Mechanics, pp. 719-728, 1962.

[4] Ben-Artzi, M., and Falcovitz, J., "A High-Resolution Upwind Scheme for Qausi 1-D Flows", INRIA Workshop on ***Numerical Methods for Solving the Euler Equations of Fluid Dynamics***, F. Angrand and R. Glowinski (editors), Paris, Dec. 1983, SIAM Publication, Philadelphia, 1985.

[5] Ben-Artzi, M., and Falcovitz, J., "A Second-Order Godunov-Type Scheme for Compressible Fluid Dynamics", Journal of Computational Physics, Vol 55, pp.1-32, 1984.

[6] Ben-Artzi, M. and Falcovitz, J., "An Upwind Second-Order Scheme for Compressible Duct Flows", SIAM Journal on Scientific and Statistical Computing, Vol 7, p.744-768, 1986.

[7] Taylor, G. I., "The Dynamics of the Combustion Products Behind Plane and Spherical Detonation Fronts in Explosives", Proc. Roy. Soc. A, Vol CC (1950), pp.235-247. Also : The Scientific Papers of Sir Geoffrey Ingram Taylor, Vol III, G. K. Batchelor, Editor, Cambridge Press, 1963.

APPENDIX A. The GRP Code

The purpose of this Appendix is to provide a concise description of the GRP code, and a listing of its CHARGE version. It is intended for users that have had prior experience in implementing schemes for solving the Euler equation of compressible flow. The theoretical background of GRP schemes constitutes the principles on which the code is founded. Some familiarity (at least) with this background, as given in References 4, 5 and 6 is indispensable to any implementation of GRP schemes. Reference 4 is recommended as an introduction. The planar GRP scheme is fully described in Reference 5, and the duct-flow GRP scheme on which the present CHARGE version is based is given in Reference 6. (In CHARGE version the flow is spherical and the "duct" area is set to X(I)**2 , but the code can handle any area variation - see subroutines CROSS and RATIO below).

In GRP schemes, second-order accuracy is achieved by considering a piecewise linear interpolation of the flow in each cell (Fig. A-1), from which second-order accurate fluxes at each cell interface are evaluated through an analysis of a local Generalized Riemann Problem (GRP). Briefly stated, the GRP goes one step further than the Riemann Problem (RP), in that it seeks (analytically) the first time-derivative of the flow that evolves as the "diaphragm" is removed from the cell interface, at the origin of the centered (X,T) wave paths of the RP solution. The major computational subroutines are CYCEUL where the integration of conservation laws is performed, RIEMAN where the local Riemann Problems are solved by Newton-Raphson iterations, MAGA where the closed-form expressions derived from the GRP analysis [6] are used to compute flow time-derivatives along the contact surface, FLUXE where all the previously computed information is used to extrapolate the fluxes to mid-time-step (T+DT/2) which constitutes a second-order accurate flux.

The plan of this Appendix is as follows. Array variables, including those which carry conserved variables (mass, momentum and energy), are described in section A.1. This is followed by descriptions of general parameters (A.2), labeled COMMON variables (A.3) and all subroutines (A.4). We conclude by giving the CHARGE version listing (A.5), which should be consulted whenever a reading of this code description is attempted.

NOTE : The present CHARGE version was implemented in a GRP code version that had been converted to treat detonation waves as chemically reactive compressible flow. However, the detonation scheme is effectively neutralized by setting QDET=0 (in NETUNM). All variables pertaining to detonation, such as arrays Z(I), DZ(I), FIMZ(I), ZMDOT(I) and labeled COMMON variables containing Z in their names, should be ignored.

A.1 Array Variables

The code GRP is organized so that all major subroutines are called with standard list of array variables which represent the integration scheme (i.e. the conservation laws), local Riemann Problem solutions and second-order accurate fluxes. Virtually all array variables are initially defined in BEGIN (initial conditions), and are subsequently updated at each time step in CYCEUL. The following list explains the meaning of these variables. Some terms used in the list are defined below.

X(I)	grid point coordinate.
U(I)	velocity in cell I.
P(I)	pressure in cell I (computed from equation of state).
RO(I)	density in cell I. This variable is time-integrated according to the law of conservation of mass. (Computed in CYCEUL).
E(I)	total energy per unit volume (including kinetic energy) in cell I. This variable is time-integrated according to the law of conservation of (total) energy. (Computed in CYCEUL).
DU(I)	velocity difference in cell I.
DP(I)	pressure difference in cell I.
DRO(I)	density difference in cell I.
DG(I)	Lagrange sound velocity difference in cell I.
DXSI(I)	the Lagrange coordinate increment defined as RO(I)*(X(I+1)-X(I)), for cell I.
MIN(I)	inactive in present version.
US(I)	velocity at the contact surface obtained after the resolution of the local discontinuity at X(I) (Riemann Problem solution). It is denoted as U^* in References 4-6.
PS(I)	pressure at the contact surface obtained after the resolution of the local discontinuity at X(I) (Riemann Problem solution). It is denoted as P^* in References 4-6.
UIDOT(I)	time derivative of US(I) along the contact surface. (This derivative is the result of the GRP analysis. It is computed in MAGA. See Ref. 5 and 6).
PIDOT(I)	time derivative of PS(I) along the contact surface. (This derivative is the result of the GRP analysis. It is computed in MAGA. See Ref. 5 and 6).
FIMZ(I)	inactive in present version.
ZMDOT(I)	inactive in present version.

TENA(I)	momentum per unit volume RO(I)*U(I) in cell I. This variable is time-integrated according to the law of conservation of momentum. (Computed in CYCEUL).
FIRO(I)	mass flux at point X(I) (second-order accurate).
FIM(I)	momentum flux at point X(I) (second-order accurate).
FIE(I)	energy flux at point X(I) (second-order accurate).
GIP(I)	the pressure term in the momentum flux. It corresponds to G(U) in References 4 and 6.
VOL(I)	volume of cell I.
Z(I)	inactive in present version.
DZ(I)	inactive in present version.

Glossary of terms used in the array variables list :

Cell I - the cell between grid points X(I) and X(I+1). All cell variables are averages per that interval.

Difference in cell I - the difference between values of variable at cell boundaries X(I+1) and X(I). Those values are obtained from "monotonized" piecewise linear distribution of each variable in each cell. (Fig. A-1).

Second-order accurate flux - the flux time-derivative at point X(I) is computed from the time-derivatives of pressure and velocity along contact surface PIDOT(I) and UIDOT(I) (in FLUXE). Then the the flux is extrapolated to the centered time point (T+DT/2), using those derivatives. This centered value is the second-order flux for integrating the conservation laws between T and T+DT.

A.2 Major Parameters

A list of major parameters indicating their meaning and the routine in which they are defined, is given below. Those parameters defined in NETUNM are the run input. There is no reading of an input file in this version of GRP code (and the only output is the printed output).

L	number of grid points + 1 (main program)
LL	L - 1 (MAIN PROGRAM)
T	time (MAIN0)
DT	time step (MAIN0)
TMAX	maximum time (when T.GE.TMAX the run is terminated) (NETUNM)
TMUD	time for which next printing will take place (NETUNM)
DTMUD	printing time step (NETUNM)
NCYC	serial number of time step (integration cycles) (MAIN0)
COLELA	switch to evaluate cell differences by Colella's method when COLELA.NE.0 (NETUNM)
KEYMON	key for monotonization scheme (just one is presently provided when COLELA.EQ.0) (NETUNM)
NCYCPR	frequency of line printing at each cycle (time step) (NETUNM)
STAB	CFL stability coefficient. Must be smaller than 1. (NETUNM)
DTBA	next time step computed from stability criterion (CYCEUL)
DTKOD	former time step (MAIN0)
KDT	index of cell where DTBA was determined (CYCEUL)

A.3 Labeled COMMON variables

Labeled COMMONs are used primarily to transmit data to and from routines that perform the major computational steps of the GRP scheme, i.e, RIEMAN, MAGA and FLUXE; these routines are called from CYCEUL. When the value of any of those variables is needed for later use, whether for updating conservation variables (RO, TENA, E), or for printing, it is stored in the appropriate array. All labeled COMMON variables are grouped under labels that indicate their role, and their names are also mnemonic. Generally, suffix L means Left and suffix R means Right. It may indicate sides either with respect to a cell interface X(I), or with respect to the contact surface which separates the Right- and Left- propagating waves in a solution to the local Riemann Problem. We indicate by INPUT variables that are computed prior to calling the subroutine, and by OUTPUT variables whose value was computed within the subroutine and constitutes the result of calling that subroutine.

COMMON /STEP0/ Parameters related to the local Riemann Problem. This is the first step in the GRP scheme.

UL, PL, ROL, CL, GL, SL - velocity, pressure, density, sound speed, Lagrange sound speed and entropy, attributed to Left side of cell interface at point X(I). (INPUT)

USTAR, PSTAR - velocity and pressure at the contact surface obtained when the local discontinuity is resolved (i.e., the solution to the local Riemann Problem). The omission of L or R suffix indicates that P and U are continuous across the contact surface. (OUTPUT)

RSTARL, CSTARL, GSTARL - density, sound speed and Lagrange sound speed on the Left side of the contact surface. (OUTPUT)

WL - Lagrange velocity of propagation of the Left-moving shock, relative to the fluid. (OUTPUT)

UW(6) - velocity of propagation of each wave front (Fig. A-3), relative to the inertial system (X). (OUTPUT)

HELEML - logical variable. If HELEML.EQ..TRUE. the Left-propagating wave is a shock. Otherwise it is a (centered) rarefaction wave. (OUTPUT)

NFLUX - integer variable. It denotes the region in the Riemann solution wave structure, which contains the point X(I) for all time. Refer to Fig. A-3 for illustration. (OUTPUT)

LAMDAL, RATEL, TEMPL, TEMPSL, ZL, ZSTARL - inactive.

COMMON /STEP1/ Parameters related to the time-derivative evaluation of the GRP scheme, performed in MAGA. The time-derivatives of P and U along the contact surface are the main result of MAGA.

DUIDT, DPIDT - time-derivatives of velocity and pressure along contact surface. (OUTPUT)

ASTARL - The directional derivative of U along the fan characteristic at the trailing characteristic of the Left rarefaction wave. It is not evaluated when the Left wave is a shock. (See References 4-6) (OUTPUT)

DGIDTL, DRIDTL - time-derivatives of Lagrange sound speed and density along the left side of the contact surface. (OUTPUT)

DSDAL - Lagrange spatial derivative of entropy on the left side of contact surface, prior to removal of the partition at X(I).

SH, RAT - the cross-section area and the x-derivative of ln(SH). They are user-defined in CROSS and RATIO respectively.

DSDASL - entropy derivative used in the special "sonic" case (i.e, when NFLUX=2 or NFLUX=5). See References 5,6 for details. (OUTPUT)

LAMDSL, DZDAL, BETACL, DZDASL - inactive.

COMMON /GRADS/ Used to transmit flow gradients (that exist in fluid prior to removal of the partition at X(I)) to MAGA.

DUDXIL, DPDXIL, DGDXIL, DRDXIL, DSDXIL - gradients of U, P, G, RO, S (with respect to Lagrange coordinate). They are computed in CYCEUL for transmission to MAGA. (INPUT)

DZDXIL - inactive.

COMMON /FI/ Used to return values of updated flux and cell-interface variables from FLUXE.

FIH1, FIH2, FIH3 - second-order flux of mass, momentum flow (just RO*U**2) and energy. They are extrapolated to Half the time step T+DT/2. (OUTPUT)

GIH - the value of P at T+DT/2

UXN, PXN, GXN, ROXN - values of U, P, G, RO extrapolated to New time T+DT, at cell-interface. They are used in CYCEUL to get tentative (pre-monotonized) new cell differences. (OUTPUT)

ZXN, FIH4, ZMDOTL, ZMDOTR - inactive.

A.4 Description of Subroutines

MAIN PROGRAM

The task of this program is to allocate array space for the NMAT arrays required by the present version of GRP code. The length of each array is L. The allocation is done by calling MAIN0. This standard calling sequence is maintained hereafter, thus facilitating modifications.

MAIN0

This subroutine functions as an overall organization routine. It can be read as a kind of flow-chart of the entire computation. First, run set-up is done by calling once to NETUNM (data) and BEGIN (initial conditions). Then a loop over time steps is begun. In each cycle the integration by one time step is performed by calling CYCEUL, and subsequently boundary conditions are implemented by calling SAFAE. Whenever T.EQ.TMUD, results are printed by calling PRINT and TMUD is updated by adding DTMUD.

NETUNM

Here data are set for a particular run. User is invited to modify this routine. There is no input file. This routine is called just once from MAIN0. Note that the detonation data section is skipped when QDET.EQ.0.

BEGIN

Initial conditions are set-up in this routine. The configuration of some nominal case is given in present version. (In CHARGE version it is the detonated spherical charge, using the Taylor self similar solution as initial conditions). User is called to modify this routine so as to generate any other desired initial configuration.

TAYLOR

The purpose of this routine, along with ancillary routines INIDAT, RUNGE and DERIV, is to compute the self-similar Taylor solution [7] of a detonated spherical charge, and implement it as initial conditions for the GRP computation of the ensuing expansion. TAYLOR is called once by BEGIN.

The core of the solution is the numerical (Runge-Kutta) integration of two coupled ordinary differential equations. The integration variable is PSI. (The flow velocity normalized by DCJ is given

by U = EXP(-PSI)). The two dependent variables are X - the normalized radial coordinate (X = 1 at the sphere boundary), and C - the normalized speed of sound. The integration is carried out by calling RUNGE, which in turn calls DERIV for the evaluation of derivatives. Data for the TAYLOR computation is set up by calling (just once) INIDAT.

The initial conditions needed in BEGIN are values of mass, momentum and (total) energy per cell. These are most accurately computed by spatially integrating the Taylor solution, resulting in lumped mass, momentum and energy per cell, which are then divided by the cell volume. This refinement is significant since gradients are high near the charge boundary (X = 1). A total mass and energy check for the entire sphere is performed and printed.

INIDAT, RUNGE, DERIV

Subroutines used only in conjunction with the Taylor initial conditions setup. See TAYLOR above.

RATIO, CROSS

User-defined routines. If A(X) is the duct cross-section area, then CROSS(X) = A(X) and RATIO(X) = D[ln(A(X))]/DX .

CYCEUL

This is the central computation routine. All major stages of the GRP scheme are performed by calling specific subroutines from CYCEUL. Then RO(I), TENA(I) and E(I) are updated to new time T + DT by solving the appropriate conservation laws in CYCEUL.

The first loop (DO 1) performs a set of preparatory steps as follows :

(a) CALL RIEMAN - Solving the local Riemann Problem at each X(I).

(b) CALL MAGA - Solving the local Generalized Riemann Problem at each X(I).

(c) CALL FLUXE - Computing second-order fluxes at X(I).

(d) Evaluation of cell-interface finite differences DU(I), DP(I), DRO(I) in each cell. These will be used at the future time step (after monotonization) for piecewise-linear interpolation of the flow in each cell. (See definition of DUDXIL, DPDXIL,..., just preceding the call to MAGA in this loop).

Note that in present CHARGE version additional computation of PRESS, PULSE1,..., PULSE4 has been added. It is just informative and does not interfere in any way with the execution of the

GRP scheme. The purpose of this computation is to monitor the numerical solution and to observe the accuracy within which the asymptotic value of the momentum integral Z (Eq. 2-9 above) is approached.

In the second loop (DO 2), the integration of the three conservation laws is performed, using second-order fluxes that had been computed in loop 1. Flow variables such as P(I) and U(I) are computed in this loop from the conserved variables. The cycle computation is concluded by calling BDOK1 for monotonization of DU(I), DP(I) and DRO(I).

SAFAE

In this routine user-defined boundary conditions are implemented. Present version (CHARGE) contains rigid wall at the center of the sphere X(2)=0, and an "open boundary" at the outer computational zone limit X(L). The rigid wall condition is achieved by setting up a virtual antisymmetric cell next to the boundary cell, so that the solution to the local Riemann Problem will result in a non-moving contact surface (USTAR=0). The open boundary is an approximation to an ideally non-reflecting boundary. Here the virtual cell is I=L, and the flow in it is defined as a "continuation" of the flow in the adjacent last cell I=LL.

BDOK1

Here the tentative cell-interface differences DV(I) are monotonized according to neighboring average cell values V(I-1), V(I) and V(I+1). The basic idea is that the cell-interface slope DV(I) should have the same sign as the average slope V(I+1)-V(I-1). When V(I) is a local extremum DV(I) is set to zero. Also, the absolute value of DV(I) is constrained so that the jump from a cell-interface value to the adjacent average value V(I), will never be of opposite sign to DV(I).

DCOLE

When COLELA option is used (not in present CHARGE version), the pre-monotonized slopes are simply the centered difference (V(I+1)-V(I-1))/2. Note that even under this option, the monotonization routine BDOK1 is subsequently called.

PRINT

Printing of results. Reading this routine is self-explanatory. Note some features added for present CHARGE version. User is called to modify this routine to his specific needs.

SOF

Run termination when an error has been detected. ISTOP is an informative index. All printing of relevant information should be done at the calling routine prior to calling SOF. Note that the run is ended in SOF by deliberately causing a system error of computing SQRT(-1). This is done in order to trigger printing of the sequence of calling routines by the operating system.

RIEMAN

Here a single Riemann Problem (RP) is solved by calling RIEMAN from CYCEUL. Referring to Fig. A-2, the RP is solved by finding the point of intersection (USTAR,PSTAR) of Left-propagating and Right-propagating shock/rarefaction adiabats in the (U,P) plane. Prior to the actual computation, the qualitative wave structure is determined. It is characterized by the index NCASE as follows :

NCASE = 1 - Left wave is rarefaction, Right wave is shock.

NCASE = 2 - Both waves are shock.

NCASE = 3 - Left wave is shock, Right wave is rarefaction.

NCASE = 4 - Both waves are rarefaction.

The computation of (USTAR,PSTAR) is coded separately for each case. Newton-Raphson iteration is employed, the first guess being the intersection of the Left and Right rarefaction branches (or their extrapolations), which is done in closed-form. Since in a smooth flow this guess is close to the exact (USTAR,PSTAR), little extra CPU effort is spent on subsequent Newton-Raphson iterations. These are truly needed only in regions of shock wave computation.

The computation in RIEMAN is concluded by computing UW(1),...,UW(5) (UW(6) = infinity). From these wave speeds, the flux index NFLUX that denotes the location of the X-axis on the (X,T) wave diagram of the RP solution (Fig. A-3), is evaluated. It is later needed in subroutine FLUXE.

MAGA

The major purpose of this routine is to compute DUIDT and DPIDT along the contact surface of the RP solution. Since U and P are continuous across the contact, so are their time-derivatives along the contact. Thus, DUIDT and DPIDT are solved from a set of two linear equations. The coefficients of each equation are determined by GRP analysis of the wave on one side. See References 4-6 (particularly Ref. 6) for details.

FLUXE

The major task of this routine is to compute second-order fluxes. This is done in two phases. The first phase is up to statement 9 CONTINUE, where using NFLUX the X-suffixed values of flow variables and their time-derivatives are defined. An X-suffix means that the variable or its time-derivative are related to the line X = X(I) on the (X,T) wave diagram (Fig. A-3). In the second phase, these variables and their time-derivatives are used to extend fluxes at X(I) to Half-time-step (hence the suffix H), i.e. T + DT/2. It is these fluxes which are the second-order accurate fluxes for the integration of the conservation laws from T to T + DT. Also, cell-interface flow variables (suffix N) are extended to New time level T + DT. These are later used in defining cell differences DU(I), DP(I) and DRO(I) in CYCEUL.

A.5 Listing of GRP Code

```
C$OPTIONS LIST
      IMPLICIT REAL*8(A-H,O-Z,$)                         CHARGE VERSION
C  PROGRAM GRP - GENERALIZED RIEMANN PROBLEM.
C  EXPANSION OF A DETONATED SPHERICAL CHARGE IN VACUUM.
C  INITIAL CONDITIONS FROM TAYLOR'S SELF SIMILAR SOLUTION.
      COMMON B(102,26)
     1      ,ENDB
      COMMON /AB/A(50)
      EQUIVALENCE (L,A(1)),(LL,A(2)),(T,A(3)),(DT,A(4)),(TMAX,A(5)),
     1            (TMUD,A(6)),(DTMUD,A(7)),(JOB,A(8)),(NERI,A(9)),
     2            (JJJ,A(10)),(KEYMON,A(11)),(NCYC,A(12))
      EQUIVALENCE (COLELA,A(13))
      EQUIVALENCE (LAGEUL,A(14))
      EQUIVALENCE (UGAL,A(15))
      EQUIVALENCE (KEYEK,A(16))
      EQUIVALENCE (NCYCPR,A(17))
      EQUIVALENCE (STAB,A(18)),(DTBA,A(19)),(DTKOD,A(20)),(KDT,A(21))
      COMMON /MONIT/CASEAV(4),NC14(4),NF16(6),
     1                   NMONU(4),NMONP(4),NMONRO(4),NMONZ(4)
      DIMENSION NZERO(26)
      EQUIVALENCE (NZERO(1),NC14(1))
      COMMON/PULS/PRESS(10),PULSE1(10),PULSE2(10),PULSE3(10),PULSE4(10)
C*********************************************************************
      DO 20 N=1,26
 20   NZERO(N)=0
      DO 21 N=1,4
 21   CASEAV(N)=0.
      DO 31 N=1,10
      PRESS(N)=0.
      PULSE1(N)=0.
      PULSE2(N)=0.
      PULSE3(N)=0.
      PULSE4(N)=0.
 31   CONTINUE
      NMAT=26
C     L=(LOCF(ENDB)-LOCF(B(1,1)))/NMAT
      L=102
      LL=L-1
      NN=NMAT*L
      DO 1 I=1,L
      DO 1 II=1,NMAT
 1    B(I,II)=0.
      CALL MAINO(L,B(1, 1),B(1, 2),B(1, 3),B(1, 4),B(1, 5),
     1             B(1, 6),B(1, 7),B(1, 8),B(1, 9),B(1,10),
     2             B(1,11),B(1,12),B(1,13),B(1,14),B(1,15),
     3             B(1,16),B(1,17),B(1,18),B(1,19),B(1,20),
     4             B(1,21),B(1,22),B(1,23),B(1,24),B(1,25),
     5             B(1,26))
      STOP
      END
```

```
      SUBROUTINE MAINO                                           MAINO
     1             (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,
     2              US,PS,UIDOT,PIDOT,
     *              FIMZ,ZMDOT,
     3              TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION X(L),U(L),P(L),RO(L),G(L),E(L),DU(L),DP(L),DRO(L),
     1           DG(L),DXSI(L),MIN(L),
     2           US(L),PS(L),UIDOT(L),PIDOT(L)
     3          ,TENA(L),FIRO(L),FIM(L),FIE(L)
     4          ,GIP(L),VOL(L),Z(L),DZ(L)
     5          ,FIMZ(L),ZMDOT(L)
      COMMON /AB/A(50)
      EQUIVALENCE (LL,A(2)),(T,A(3)),(DT,A(4)),(TMAX,A(5)),
     1            (TMUD,A(6)),(DTMUD,A(7)),(JOB,A(8)),(NERI,A(9)),
     2            (JJJ,A(10)),(KEYMON,A(11)),(NCYC,A(12))
      EQUIVALENCE (LAGEUL,A(14))
      EQUIVALENCE (NCYCPR,A(17))
      EQUIVALENCE (STAB,A(18)),(DTBA,A(19)),(DTKOD,A(20)),(KDT,A(21))
      COMMON /TOT/AMTOT,ETOT,EKTOT,EPTOT,TENTOT
C*********************************************************************
      T=0.
```

```
      NCYC=0
      JJJ=0
      CALL NETUNM
      DELT=DT
      CALL BEGIN
     1                 (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,
     2                  US,PS,UIDOT,PIDOT,
     *                 FIMZ,ZMDOT,
     3                  TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)
      CALL SAFAE
     1                 (L,X,U,F,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,
     2                  US,PS,UIDOT,PIDOT,
     *                 FIMZ,ZMDOT,
     3                  TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)
    1 NCYC=NCYC+1
C TIME STEP CONTROL.
      DT=DTBA
      IF(DT.GT.1.1D0*DTKOD.AND.DTKOD.NE.0.) DT=1.1D0*DTKOD
      IF(NCYC.EQ.2) DT=DT/10.D0
      IF (NCYC.EQ.1) DT=0.
      IF(DT.EQ.0.) GO TO 11
      NHAD=((TMUD-T)/DT-1.D-10)
      IF(NHAD.GE.10) GO TO 11
      DT=(TMUD-T)/DFLOAT(NHAD+1)
   11 CONTINUE
      T=T+DT
      IF((NCYC/NCYCPR)*NCYCPR.NE.NCYC.AND.NCYC.GT.NCYCPR) GO TO 33
      PRINT 10, NCYC,T,DT,KDT
   10 FORMAT(1X,'NCYC=',I4,3X,'T=',D11.4,3X,'DT=',D11.4,3X,'KDT=',I4)
   33 CONTINUE
      DTBA=DTMUD
      KDT=0
      NERI=1
      IF (DABS(T-TMUD).LT.1.D-8) NERI=0
      CALL CYCEUL
     1                 (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,
     2                  US,PS,UIDOT,PIDOT,
     *                 FIMZ,ZMDOT,
     3                  TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)
      CALL SAFAE
     1                 (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,
     2                  US,PS,UIDOT,PIDOT,
     *                 FIMZ,ZMDOT,
     3                  TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)
      IF (NERI.NE.0) GO TO 2
      CALL PRINT
     1                 (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,
     2                  US,PS,UIDOT,PIDOT,
     *                 FIMZ,ZMDOT,
     3                  TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)
      IF (DABS(T-TMUD).LT.1.D-8) TMUD=TMUD+DTMUD
    2 CONTINUE
      DTKOD=DT
      IF (T.LT.TMAX-1.D-8) GO TO 1
      RETURN
      END
      SUBROUTINE NETUNM                                     NETUNM
      IMPLICIT REAL*8(A-H,O-Z,$)
      COMMON /AB/A(50)
      EQUIVALENCE (L,A(1))
      EQUIVALENCE (LL,A(2)),(T,A(3)),(DT,A(4)),(TMAX,A(5)),
     1            (TMUD,A(6)),(DTMUD,A(7)),(JOB,A(8)),(NERI,A(9)),
     2            (JJJ,A(10)),(KEYMON,A(11)),(NCYC,A(12))
      EQUIVALENCE (COLELA,A(13))
      EQUIVALENCE (LAGEUL,A(14))
      EQUIVALENCE (KEYEK,A(16))
      EQUIVALENCE (NCYCPR,A(17))
      EQUIVALENCE (STAB,A(18)),(DTBA,A(19)),(DTKOD,A(20)),(KDT,A(21))
      COMMON/DETO/QDET,PCJDET,RCJDET,UCJDET,DCJDET,PODET,ROODET,
     1            RATE,TEMPC
      COMMON/DIFFUS/U2,P2,RO2,ARW
      COMMON /DRAW/GODELX,GODELY,UMIN,UMAX,PMIN,PMAX,ROMIN,ROMAX
```

```
     1              ,XMIN,XMAX,SMIN,SMAX,IVERSA
      COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11
     1          ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23
     2          ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35
      REAL*8 NG,MU2
      NAMELIST /IN/LIN,GAMA,DT,TMUD,DTMUD,TMAX,
     1              GODELX,GODELY,UMIN,UMAX,PMIN,PMAX,ROMIN,ROMAX,
     2              SMIN,SMAX,IVERSA,KEYMON,COLELA,STAB
     3             ,LAGEUL,KEYEK
     4             ,QDET
C*************************************************************************
      LIN=L
      LAGEUL=2
      NCYCPR=1
      KEYEK=1
      TMUD=0.
      DTMUD=10.D0
      TMAX=100.D0
      STAB=0.5D0
      DT=1.D-2
      KEYMON=1
      GAMA=3.D0+1.D-6
      QDET=0.04D0
      RATE=0.
      TEMPC=1.D50
      GODELX=16D0
      GODELY=20.D0
      IVERSA=100
      UMIN=0.
      UMAX= 1.D0
      PMIN=0.
      PMAX=0.5D0
      ROMIN=0.
      ROMAX=3.D0
      SMIN=0.
      SMAX=0.03D0
      COLELA=0.
C     READ IN
C
      PRINT IN
C
      GG=2.D0*GAMA/(GAMA-1.D0)
      NG=GG
   10 CONTINUE
      MU2=(GAMA-1.D0)/(GAMA+1.D0)
      G1=GAMA-1.D0
      G2=1.D0-MU2
      G3=2.D0/(3.D0*GAMA-1.D0)
      G4=(GAMA+1.D0)/2.D0
      G5=0.5D0*(3.D0*GAMA-1.D0)/(GAMA+1.D0)
      G6=(GAMA+1.D0)/(2.D0*GAMA)
      G7=2.D0/(GAMA-1.D0)
      G8=(GAMA-1.D0)/(2.D0*GAMA)
      G9=(GAMA+1.D0)/(4.D0*GAMA)
      G10=1.D0/GAMA
      G11=(GAMA+1.D0)/4.D0
      G12=GAMA/(GAMA-1.D0)
      G13=0.5D0*(GAMA-3.D0)/(GAMA+1.D0)
      G14=0.5D0*(3.D0*GAMA-5.D0)/(GAMA+1.D0)
      G15=GAMA*(3.D0*GAMA-1.D0)
      G16=(GAMA+1.D0)/(2.D0*(GAMA-1.D0))
      G17=GAMA+1.D0
      G18=GAMA*(GAMA+1.D0)/(3.D0*GAMA-1.D0)
      G19=(3.D0*GAMA-1.D0)/(GAMA+1.D0)
      G20=2.D0*(GAMA-1.D0)/(3.D0*GAMA-1.D0)**2
      G21=GAMA*(3.D0*GAMA-5.D0)/(3.D0*GAMA-1.D0)**2
      GODELX=GODELX/2.54D0
      GODELY=GODELY/2.54D0
C     CALL NAMPLT(IVERSA)
C     CALL LIMIT(1000.D0)
C     CALL PLOT(0.,0.5D0,-3)
      PODET=0.
```

```
      ROODET=0.
      PCJDET=0.
      UCJDET=0.
      DCJDET=0.
      RCJDET=0.
      IF(QDET.LE.0.) GO TO 100
C  DETONATION DATA
      QDET=0.04D0
      PODET=0.
      ROODET=1.8D0
      PCJDET=PODET-(GAMA-1.D0)*(-QDET)*ROODET+
     1   DSQRT(((GAMA-1.D0)*QDET*ROODET)**2-2.D0*MU2*GAMA*
     2   (-QDET)*PODET*ROODET)
      RCJDET=ROODET*((GAMA+1.D0)*PCJDET-PODET)/(GAMA*PCJDET)
      CCJ=DSQRT(GAMA*PCJDET/RCJDET)
      DCJDET=CCJ*RCJDET/ROODET
      UCJDET=DCJDET-CCJ
      PRINT 101
  101 FORMAT(1H1,/,1X,'DETONATION DATA'/)
      PRINT 102, QDET,GAMA,TEMPC,RATE
  102 FORMAT(/1X,'QDET,GAMA,TEMP,RATE=',4D18.8)
      PRINT 103, ROODET,PODET
  103 FORMAT(/1X,'UNBURNED STATE       ROODET,PODET=',2D18.8)
      PRINT 104, DCJDET,PCJDET,RCJDET,UCJDET
  104 FORMAT(/1X,'CJ POINT      DCJDET,PCJDET,RCJDET,UCJDET=',4D18.8)
  100 CONTINUE
      RETURN
      END
```

BEGIN

```
      SUBROUTINE BEGIN
     1              (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,
     2               US,PS,UIDOT,PIDOT,
     *              FIMZ,ZMDOT,
     3               TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION X(L),U(L),P(L),RO(L),G(L),E(L),DU(L),DP(L),DRO(L),
     1          DG(L),DXSI(L),MIN(L),
     2          US(L),PS(L),UIDOT(L),PIDOT(L)
     3         ,TENA(L),FIRO(L),FIM(L),FIE(L)
     4         ,GIP(L),VOL(L),Z(L),DZ(L)
     5         ,FIMZ(L),ZMDOT(L)
      COMMON /AB/ A(50)
      COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11
     1          ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23
     2          ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35
      REAL*8 NG,MU2
      EQUIVALENCE (LL,A(2))
      EQUIVALENCE (LAGEUL,A(14))
      EQUIVALENCE (UGAL,A(15))
      EQUIVALENCE (STAB,A(18)),(DTBA,A(19)),(DTKOD,A(20)),(KDT,A(21))
      COMMON/DETO/QDET,PCJDET,RCJDET,UCJDET,DCJDET,PODET,ROODET,
     1            RATE,TEMPC
      COMMON /DRAW/GODELX,GODELY,UMIN,UMAX,PMIN,PMAX,ROMIN,ROMAX
     1              ,XMIN,XMAX,SMIN,SMAX,IVERSA
      COMMON/GIT/ROLIM,ELIM,XGIT(200),ROGIT(200),ROUGIT(200),EGIT(200)
      COMMON /GITN/NPO
      LOGICAL CSOF
C*****************************************************************
      DTBA=0.
      DTKOD=0.
      KDT=0
      P0=1.D-9
      RHO0=1.D-7
      U0=UCJDET
      UGAL=0.
      X0=0.
      X1=50.D0
      XCHARG=10.D0
      XMIN=X0
      XMAX=X1
      DX=(X1-X0)/(L-2.D0)
      DO 1 I=2,L
      X(I)=X0+(I-2.D0)*DX
```

```
  1     CONTINUE
        X(L)=X1
        U0=U0*CROSS(XCHARG)
        DO 2 I=2,LL
        IF(I.GT.2) U(I)=U0/CROSS(X(I))
        P(I)=P0
        RO(I)=RHO0
        Z(I)=0.
        GO TO (31,32), LAGEUL
 31     CONTINUE
        E(I)=P(I)/((GAMA-1.D0)*RO(I))+0.5D0*U(I)**2+Z(I)*QDET
        GO TO 30
 32     CONTINUE
        E(I)=P(I)/(GAMA-1.D0)+0.5D0*RO(I)*U(I)**2+Z(I)*RO(I)*QDET
 30     CONTINUE
        G(I)=DSQRT(GAMA*P(I)*RO(I))
  2     CONTINUE
        DO 3 I=2,LL
        TENA(I)=RO(I)*U(I)
        VOL(I)=(X(I+1)-X(I))*(X(I+1)**2+X(I+1)*X(I)+X(I)**2)/3.D0
  3     CONTINUE
C
C  INSERT DETONATED CHARGE FLOW FIELD FROM TAYLOR'S SOLUTION.
C
        CALL TAYLOR(GAMA)
        RONORM=RCJDET
        RUNORM=RCJDET*UCJDET
        ENORM=RCJDET*DCJDET**2
        XLIM=XGIT(NP0)
        NGIT=NP0-1
        XG1=XLIM
        XG2=XGIT(NGIT)
        AROIP =ROGIT (NP0)+ROLIM*XLIM**3/3.D0
        AROUIP=ROUGIT(NP0)
        AEIP  =EGIT  (NP0)+ ELIM*XLIM**3/3.D0
        XP=X(2)/XCHARG
        DO 100 I=2,LL
        IP=I+1
        XI=XP
        AROI  =AROIP
        AROUI=AROUIP
        AEI   =AEIP
        XP=X(IP)/XCHARG
        IF(DABS(XP-1.D0).LT.1.D-10) XP=1.D0
        CSOF=(XP.GE.1.D0)
        IF(XP.GE.XLIM) GO TO 101
C  UNIFORM FLOW REGION
        DELVOL=(XLIM-XP)*(XLIM**2+XLIM*XP+XP**2)/3.D0
        AROIP =ROGIT (NP0)+ROLIM*DELVOL
        AROUIP=ROUGIT(NP0)
        AEIP  =EGIT  (NP0)+ ELIM*DELVOL
        GO TO 102
 101    CONTINUE
C  NON UNIFORM FLOW REGION.
        IF(.NOT.CSOF) GO TO 104
C  LAST POINT.  (THIS IS THE DETONATION FRONT POINT  X=1).
        AROIP= 0.
        AROUIP=0.
        AEIP=  0.
        GO TO 102
 104    CONTINUE
        IF(XP.LE.XG2) GO TO 103
        NGIT=NGIT-1
        IF(NGIT.LE.0) CALL SOF('BEGIN 104. NGIT.LE.0.')
        XG1=XG2
        XG2=XGIT(NGIT)
        GO TO 104
 103    CONTINUE
        FRAC=(XP-XG1)/(XG2-XG1)
        IF(FRAC.LT.0.   ) CALL SOF('BEGIN 103.  FRAC.LT.0.')
        IF(FRAC.GT.1.D0) CALL SOF('BEGIN 103.  FRAC.GT.1.')
        AROIP =(1.D0-FRAC)*ROGIT (NGIT+1)+FRAC*ROGIT (NGIT)
```

```
      AROUIP=(1.D0-FRAC)*ROUGIT(NGIT+1)+FRAC*ROUGIT(NGIT)
      AEIP  =(1.D0-FRAC)*EGIT  (NGIT+1)+FRAC*EGIT  (NGIT)
 102  CONTINUE
C  COMPUTE MASS, MOMENTUM AND ENERGY DENSITIES.
      IF(XP.LE.XLIM) GO TO 105
C  CONSERVATION-FORM DEFINITION OF MASS, MOMENTUM AND ENERGY DENSITY.
      DVOL=(XP-XI)*(XP**2+XP*XI+XI**2)/3.D0
      RO  (I)=RONORM*(AROI - AROIP)/DVOL
      TENA(I)=RUNORM*(AROUI-AROUIP)/DVOL
      E   (I)=ENORM *(AEI  -  AEIP)/DVOL
      GO TO 106
 105  CONTINUE
C  UNIFORM FLOW REGION
      RO  (I)=RONORM*ROLIM
      TENA(I)=0.
      E   (I)=ENORM * ELIM
 106  CONTINUE
      U(I)=TENA(I)/RO(I)
      P(I)=(GAMA-1.D0)*(E(I)-0.5D0*RO(I)*U(I)**2)
      PRINT 111,I,CSOF,U(I),P(I),RO(I),E(I)
 111  FORMAT(/1X,'I,CSOF,U,P,RO,E=',I4,L3,4D14.4)
      IF(CSOF) GO TO 109
 100  CONTINUE
 109  CONTINUE
      DO 4 I=2,LL
      DXSI(I)=(X(I+1)-X(I))*RO(I)
 4    CONTINUE
      RETURN
      END
      SUBROUTINE TAYLOR(GAMA)                          TAYLOR
      IMPLICIT REAL*8(A-H,O-Z,$)
C
C  TAYLOR -- SELF SIMILAR SPHERICAL DETONATION (CJ) FLOW FIELD
C
      COMMON /GGGG/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10
      COMMON /PAR/RHO0,Q0,ROCJ,DCJ,UCJ,PCJ,DPSI,PSIMAX,C0,U0
      COMMON /GITN/NP0
      COMMON/GIT/ROLIM,ELIM,XGIT(200),ROGIT(200),ROUGIT(200),EGIT(200)
C***********************************************************************
      G=GAMA
      PRINT 101
 101  FORMAT('1')
      PRINT 110
 110  FORMAT(1X,'G. I. TAYLOR SOLUTION.  N,PSI,U,C,X/AM,AT,AE='//)
      CALL INIDAT
      X=1.D0
      Y=0.
      U=U0
      C=C0
      AM=0.
      AT=0.
      AE=0.
      PSI=-DLOG(U)
      DO 1 N=1,NP0
      XGIT  (N)=X
      ROGIT (N)=AM
      ROUGIT(N)=AT
      EGIT  (N)=AE
      PRINT 11, N,PSI,U,C,X,AM,AT,AE
 11   FORMAT(1X,I4,4D14.5/5X,3D14.5)
      CALL RUNGE(N,PSI,X,C,AM,AT,AE,PSIN,XN,CN,AMN,ATN,AEN)
      PSI=PSIN
      U=DEXP(-PSI)
      X=XN
      C=CN
      AM=AMN
      AT=ATN
      AE=AEN
 1    CONTINUE
      ROLIM=(C/C0)**G3
      ELIM=G5*(C/C0)**G4
      AM0=AM+(C/C0)**G3*X**3/3.D0
```

```
      AM0=AM0*3.D0*(G+1.D0)/G
      AE0=AE+(G5*(C/C0)**G4+0.5D0*(C/C0)**G3*U**2)*X**3/3.D0
      AE0=AE0*6.D0*(G+1.D0)*(G**2-1.D0)/G
      PRINT 22,AM0,AE0
   22 FORMAT(///1X,'MASS AND ENERGY CHECK (SHOULD BE 1.)'//
     1          1X,'M0=',D17.8,5X,'E0=',D17.8//)
      RETURN
      END
      SUBROUTINE INIDAT                                    INIDAT
      IMPLICIT REAL*8(A-H,O-Z,$)
      COMMON /GGGG/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10
      COMMON /PAR/RHO0,Q0,ROCJ,DCJ,UCJ,PCJ,DPSI,PSIMAX,C0,U0
      COMMON /GITN/NP0
C**********************************************************************
      NP0=200
      PSIMAX=10.D0
      U0=1.D0/(G+1.D0)
      C0=1.D0-U0
      DPSI=PSIMAX/DFLOAT(NP0)
      G1=G-1.D0
      G2=G1/2.D0
      G3=2.D0/(G-1.D0)
      G4=2.D0*G/(G-1.D0)
      G5=G/((G+1.D0)**2*(G-1.D0))
      RETURN
      END                                                     RUNGE
      SUBROUTINE RUNGE(N,PSI,X,C,AM,AT,AE,PSIN,XN,CN,AMN,ATN,AEN)
      IMPLICIT REAL*8(A-H,O-Z,$)
      COMMON /GGGG/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10
      COMMON /PAR/RHO0,Q0,ROCJ,DCJ,UCJ,PCJ,DPSI,PSIMAX,C0,U0
      COMMON /GITN/NP0
C**********************************************************************
      H=DPSI
      H2=H/2.D0
      H6=H/6.D0
      CALL DERIV(PSI,X,C,AM,AT,AE,
     1                   DXDP1,DCDP1,DMDP1,DTDP1,DEDP1)
      CALL DERIV(PSI+H2,X+H2*DXDP1,C+H2*DCDP1,AM,AT,AE,
     1                   DXDP2,DCDP2,DMDP2,DTDP2,DEDP2)
      CALL DERIV(PSI+H2,X+H2*DXDP2,C+H2*DCDP2,AM,AT,AE,
     1                   DXDP3,DCDP3,DMDP3,DTDP3,DEDP3)
      CALL DERIV(PSI+H,X+H*DXDP3,C+H*DCDP3,AM,AT,AE,
     1                  DXDP4,DCDP4,DMDP4,DTDP4,DEDP4)
      PSIN=PSI+H
      XN=X+H6*(DXDP1+2.D0*(DXDP2+DXDP3)+DXDP4)
      CN=C+H6*(DCDP1+2.D0*(DCDP2+DCDP3)+DCDP4)
      AMN=AM+H6*(DMDP1+2.D0*(DMDP2+DMDP3)+DMDP4)
      ATN=AT+H6*(DTDP1+2.D0*(DTDP2+DTDP3)+DTDP4)
      AEN=AE+H6*(DEDP1+2.D0*(DEDP2+DEDP3)+DEDP4)
      RETURN
      END                                                     DERIV
      SUBROUTINE DERIV(PSI,X,C,AM,AT,AE,DXDP,DCDP,DMDP,DTDP,DEDP)
      IMPLICIT REAL*8(A-H,O-Z,$)
      COMMON /GGGG/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10
      COMMON /PAR/RHO0,Q0,ROCJ,DCJ,UCJ,PCJ,DPSI,PSIMAX,C0,U0
      COMMON /GITN/NP0
C**********************************************************************
      U=DEXP(-PSI)
      DXDP=0.5D0*X*(C-U+X)*(C+U-X)/C**2
      DCDP=-G2*U*(X-U)/C
      DMDP=-(C/C0)**G3*X**2*DXDP
      DTDP=DMDP*U
      DEDP=-(G5*(C/C0)**G4+0.5D0*(C/C0)**G3*U**2)*X**2*DXDP
      RETURN
      END
      DOUBLE PRECISION FUNCTION RATIO(X)                   RATIO
      IMPLICIT REAL*8(A-H,O-Z,$)
C**********************************************************************
      RATIO=0.
      IF(X.LE.1.D-8)RETURN
      RATIO=2.D0/X
      RETURN
```

FILE: CHARGEP FORTRAN A1

```
      END                                                               CHA0512
      DOUBLE PRECISION FUNCTION CROSS(X)                CROSS           CHA0513
      IMPLICIT REAL*8(A-H,O-Z,$)                                        CHA0514
C***********************************************************************CHA0515
C     CROSS=1.D0                                                        CHA0516
      CROSS=X**2                                                        CHA0517
      RETURN                                                            CHA0518
      END                                                               CHA0519
      SUBROUTINE CYCEUL                                      CYCEUL     CHA0520
     1               (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,             CHA0521
     2                US,PS,UIDOT,PIDOT,                                CHA0522
     *               FIMZ,ZMDOT,                                        CHA0523
     3                TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)                   CHA0524
      IMPLICIT REAL*8(A-H,O-Z,$)                                        CHA0525
      DIMENSION X(L),U(L),P(L),RO(L),G(L),E(L),DU(L),DP(L),DRO(L),      CHA0526
     1          DG(L),DXSI(L),MIN(L),                                   CHA0527
     2          US(L),PS(L),UIDOT(L),PIDOT(L)                           CHA0528
     3         ,TENA(L),FIRO(L),FIM(L),FIE(L)                           CHA0529
     4         ,GIP(L),VOL(L),Z(L),DZ(L)                                CHA0530
     5         ,FIMZ(L),ZMDOT(L)                                        CHA0531
      COMMON /AB/A(50)                                                  CHA0532
      EQUIVALENCE (LL,A(2)),(T,A(3)),(DT,A(4)),(COLELA,A(13))           CHA0533
      EQUIVALENCE (KEYEK,A(16))                                         CHA0534
      EQUIVALENCE (STAB,A(18)),(DTBA,A(19)),(DTKOD,A(20)),(KDT,A(21))   CHA0535
      COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11        CHA0536
     1          ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23        CHA0537
     2          ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35        CHA0538
      REAL*8 NG,MU2                                                     CHA0539
      COMMON /TOT/AMTOT,ETOT,EKTOT,EPTOT,TENTOT                         CHA0540
      COMMON /AZOV/ISAFA,NORIMN,USAF,PSAF,ROSAF,GSAF,ESAF,DPSAF         CHA0541
     1             ,DXSIL,DXSIR                                         CHA0542
      LOGICAL NORIMN                                                    CHA0543
      COMMON /STEP0/UL,PL,ROL,GL,UR,PR,ROR,GR,USTAR,PSTAR,              CHA0544
     1               RSTARL,RSTARR,GSTARL,GSTARR,                       CHA0545
     2               CL,CR,CSTARL,CSTARR,SL,SR,WL,WR,UW(6)              CHA0546
     3              ,LAMDAL,LAMDAR,RATEL,RATER,TEMPL,TEMPR,TEMPSL,TEMPSR CHA0547
     4              ,ZL,ZR,ZSTARL,ZSTARR,NFLUX,HELEML,HELEMR            CHA0548
      REAL*8 LAMDAL,LAMDAR                                              CHA0549
      LOGICAL HELEML,HELEMR                                             CHA0550
      COMMON /STEP1/DUIDT,DPIDT,DGIDTL,DGIDTR,DRIDTL,DRIDTR             CHA0551
     2  ,ASTARL,ASTARR,LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR            CHA0552
     3   ,RAT,SH                                                        CHA0553
     4   ,BETACL,BETACR,DSDASL,DSDASR,DZDASL,DZDASR                     CHA0554
      REAL*8 LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR                      CHA0555
      COMMON /GRADS/DUDXIL,DPDXIL,DGDXIL,DRDXIL,DZDXIL,DSDXIL,          CHA0556
     1              DUDXIR,DPDXIR,DGDXIR,DRDXIR,DZDXIR,DSDXIR           CHA0557
      COMMON /FI/FIH1,FIH2,FIH3,UXN,PXN,GXN,ROXN,ZXN                    CHA0558
     1   ,GIH                                                           CHA0559
     2   ,FIH4,ZMDOTL,ZMDOTR                                            CHA0560
      COMMON/DETO/QDET,PCJDET,RCJDET,UCJDET,DCJDET,PODET,ROODET,        CHA0561
     1            RATE,TEMPC                                            CHA0562
      COMMON/PULS/PRESS(10),PULSE1(10),PULSE2(10),PULSE3(10),PULSE4(10) CHA0563
      DATA ERRP/0.D0/                                                   CHA0564
      DATA NERRP/0/                                                     CHA0565
C     DATA KOTZ/7777777777B/                                            CHA0566
C***********************************************************************CHA0567
      DT2=DT/2.D0                                                       CHA0568
      UXN=0.                                                            CHA0573
      PXN=0.                                                            CHA0574
      ROXN=0.                                                           CHA0575
      ZXN=0.                                                            CHA0576
      DO 1 I=2,L                                                        CHA0577
      IM=I-1                                                            CHA0578
      UXNM=UXN                                                          CHA0579
      PXNM=PXN                                                          CHA0580
      ROXNM=ROXN                                                        CHA0581
      ZXNM=ZXN                                                          CHA0582
      UL=U(IM)+0.5D0*DU(IM)                                             CHA0583
      PL=P(IM)+0.5D0*DP(IM)                                             CHA0584
      ROL=RO(IM)+0.5D0*DRO(IM)                                          CHA0585
      GL=DSQRT(GAMA*PL*ROL)                                             CHA0586
      CL=GL/ROL                                                         CHA0587
```

```
      ZL=Z(IM)+0.5D0*DZ(IM)
      IF(ZL.GT.1.D0) ZL=1.D0
      IF(ZL.LT.0.  ) ZL=0.
      SL=PL/(G1*ROL**GAMA)
      TEMPL=PL/ROL
      UR=U(I)-0.5D0*DU(I)
      PR=P(I)-0.5D0*DP(I)
      ROR=RO(I)-0.5D0*DRO(I)
      GR=DSQRT(GAMA*PR*ROR)
      CR=GR/ROR
      ZR=Z(I)-0.5D0*DZ(I)
      IF(ZR.GT.1.D0) ZR=1.D0
      IF(ZR.LT.0.  ) ZR=0.
      SR=PR/(G1*ROR**GAMA)
      TEMPR=PR/ROR
C
      CALL RIEMAN(L,I,MIN)
C
      DUDXIL=DU(IM)/DXSI(IM)
      DPDXIL=DP(IM)/DXSI(IM)
      DRDXIL=DRO(IM)/DXSI(IM)
      DGDXIL=0.5D0*GL*(DPDXIL/PL+DRDXIL/ROL)
      DZDXIL=DZ(IM)/DXSI(IM)
      DSDXIL=SL*(DPDXIL/PL-GAMA*DRDXIL/ROL)
      DUDXIR=DU(I)/DXSI(I)
      DPDXIR=DP(I)/DXSI(I)
      DRDXIR=DRO(I)/DXSI(I)
      DGDXIR=0.5D0*GR*(DPDXIR/PR+DRDXIR/ROR)
      DZDXIR=DZ(I)/DXSI(I)
      DSDXIR=SR*(DPDXIR/PR-GAMA*DRDXIR/ROR)
      SH=CROSS(X(I))
      RAT=RATIO(X(I))
C
      CALL MAGA(L,I,MIN)
C
      US(I)=USTAR
      PS(I)=PSTAR
      UIDOT(I)=DUIDT
      PIDOT(I)=DPIDT
C
      CALL FLUXE(L,I,MIN)
C
      FIRO(I)=FIH1
      FIM (I)=FIH2
      FIE (I)=FIH3
      FIMZ(I)=FIH4
      GIP(I)=GIH
      DU(IM)=UXN-UXNM
      DP(IM)=PXN-PXNM
      DRO(IM)=ROXN-ROXNM
      DZ(IM)=ZXN-ZXNM
C  STATIONS OUTPUT
      IF((I-42)*(I-62)*(I-82)*(I-102).NE.0) GO TO 1
      NPU=0
      IF(I.EQ.42) NPU=1
      IF(I.EQ.62) NPU=2
      IF(I.EQ.82) NPU=3
      IF(I.EQ.102)NPU=4
      IF(NPU.EQ.0) CALL SOF('FLUXE 90.  NPU.EQ.0')
      PRESS(NPU)=GIH+FIH2
      PULSE1(NPU)=PULSE1(NPU)+DT*GIH
      PULSE2(NPU)=PULSE2(NPU)+DT*(GIH+FIH2)
      PULSE3(NPU)=PULSE3(NPU)+DT*FIH1*CROSS(X(I))
      PULSE4(NPU)=PULSE4(NPU)+DT*FIH2*CROSS(X(I))
    1 CONTINUE
C
      AMTOT=0.
      ETOT=0.
      EKTOT=0.
      EPTOT=0.
      TENTOT=0.
      FI1=FIRO(2)
```

```
      FI2=FIM (2)
      FI3=FIE (2)
      FI4=FIMZ(2)
      GI2=GIP(2)
      SH=CROSS(X(2))
      DO 2 I=2,LL
      IP=I+1
      FIM1=FI1
      FIM2=FI2
      FIM3=FI3
      FIM4=FI4
      GIM2=GI2
      SHM=SH
      FI1=FIRO(IP)
      FI2=FIM (IP)
      FI3=FIE (IP)
      FI4=FIMZ(IP)
      GI2=GIP (IP)
      SH=CROSS(X(IP))
      DVOL=VOL(I)
      ROOLD=RO(I)
      POLD=P(I)
      EOLD=E(I)
      UOLD=U(I)
      ZOLD=Z(I)
      ZKODM=ZOLD*ROOLD
      TOLD=POLD/ROOLD
      DX=X(IP)-X(I)
      DTVOL=DT/DVOL
C
      RO(I)=RO(I)-DTVOL*(SH*FI1-SHM*FIM1)
      TENA(I)=TENA(I)-DTVOL*(SH*FI2-SHM*FIM2)-(DT/DX)*(GI2-GIM2)
      E(I)=E(I)-DTVOL*(SH*FI3-SHM*FIM3)
      U(I)=TENA(I)/RO(I)
      Z(I)=(ZKODM-DTVOL*(SH*FI4-SHM*FIM4))/RO(I)
      IF(Z(I).GT.1.D0) Z(I)=1.D0
      IF(Z(I).LT.0.) Z(I)=0.
C
      UAV=U(I)
      ROAV=RO(I)
      EP=E(I)-0.5D0*ROAV*UAV**2
      IF(EP.GT.0.) GO TO 291
      NERRP=NERRP+1
      ERRP=ERRP+(1.D-8-EP)*DVOL
      IF(ERRP.GT.0.24D0) GO TO 291
      EP=1.D-8
  291 CONTINUE
      IF(EP.LE.0.) GO TO 7001
      P(I)=G1*EP
      G(I)=DSQRT(GAMA*P(I)*RO(I))
C
      UPC=DABS(U(I))+G(I)/RO(I)
      DTI=STAB*DX/UPC
      IF(DTI.GT.DTBA) GO TO 29
      DTBA=DTI
      KDT=I
   29 CONTINUE
      DXSI(I)=RO(I)*DX
      ETOT=ETOT+E(I)*DVOL
      EPTOT=EPTOT+EP*DVOL
      AMTOT=AMTOT+RO(I)*DVOL
      TENTOT=TENTOT+TENA(I)*DVOL
    2 CONTINUE
      EKTOT=ETOT-EPTOT
C
      IF(COLELA.EQ.0.) GO TO 200
      CALL DCOLE(L,X,U ,DU ,MIN,1)
      CALL DCOLE(L,X,P,DP,MIN,2)
      CALL DCOLE(L,X,RO,DRO,MIN,3)
      CALL DCOLE(L,X,Z,DZ,MIN,4)
  200 CONTINUE
      CALL BDOK1(L,X,U ,DU ,MIN,1)
```

```
      CALL BDOK1(L,X,P,DP,MIN,2)
      CALL BDOK1(L,X,RO,DRO,MIN,3)
      CALL BDOK1(L,X,Z,DZ,MIN,4)
      PRINT 901,(NN,PRESS(NN),PULSE1(NN),PULSE2(NN),
     1           PULSE3(NN)/AMTOT,PULSE4(NN)/TENTOT,NN=1,4)
  901 FORMAT(1X,2('/',I3,5D11.3,'/')/)
      IF(DABS(T-A(5)).LT.1.D-6) PRINT 911,NERRP,ERRP
  911 FORMAT(//1X,'NERRP,ERRP=',I5,D15.5/)
      RETURN
 7001 CONTINUE
      PRINT 7101, I,ROAV,UAV,DRO(I),DU(I),E(I),EP,ZNEW,ZNEW-1.D0,EPI
 7101 FORMAT(//1X,'FROM CYCEUL.  NEGATIVE EP.  IN CELL  I=',I6//
     1          1X,'ROAV,UAV,DRO(I),DU(I)=',4D18.8//
     2          1X,'E(I),EP,ZNEW,ZNEW-1,EPI=',5D14.6//)
      CALL PRINT
     1             (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,
     2              US,PS,UIDOT,PIDOT,
     *             FIMZ,ZMDOT,
     3              TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)
      CALL SOF('CYCEUL 7001, NEGATIVE EP')
      RETURN
      END
      SUBROUTINE SAFAE                                         SAFAE
     1             (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,
     2              US,PS,UIDOT,PIDOT,
     *             FIMZ,ZMDOT,
     3              TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION X(L),U(L),P(L),RO(L),G(L),E(L),DU(L),DP(L),DRO(L),
     1          DG(L),DXSI(L),MIN(L),
     2          US(L),PS(L),UIDOT(L),PIDOT(L)
     3         ,TENA(L),FIRO(L),FIM(L),FIE(L)
     4         ,GIP(L),VOL(L),Z(L),DZ(L)
     5         ,FIMZ(L),ZMDOT(L)
      COMMON /AB/A(50)
      EQUIVALENCE (LL,A(2)),(T,A(3)),(DT,A(4)),(NCYC,A(12))
      EQUIVALENCE (UGAL,A(15))
      COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11
     1          ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23
     2          ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35
      REAL*8 NG,MU2
      COMMON/DETO/QDET,PCJDET,RCJDET,UCJDET,DCJDET,PODET,ROODET,
     1            RATE,TEMPC
      COMMON/DIFFUS/U2,P2,RO2,ARW
C**************************************************************************
C
C  RIGID B.C. AT I=2
C
      U(1)=-U(2)
      P(1)=P(2)
      G(1)=G(2)
      RO(1)=RO(2)
      Z(1)=Z(2)
      DU(1)=DU(2)
      DP(1)=-DP(2)
      DG(1)=-DG(2)
      DRO(1)=-DRO(2)
      DXSI(1)=DXSI(2)
C
C  OUTFLOW B.C. AT I=L
C
      U(L)=U(LL)+DU(LL)/2.D0
      P(L)=P(LL)+DP(LL)/2.D0
      RO(L)=RO(LL)+DRO(LL)/2.D0
      G(L)=G(LL)+DG(LL)/2.D0
      Z(L)=Z(LL)+DZ(LL)/2.D0
      DU(L)=0.
      DP(L)=0.
      DG(L)=0.
      DRO(L)=0.
      DZ(L)=0.
      DXSI(L)=DXSI(LL)
```

```
C                                                                       CHA0804
      RETURN                                                            CHA0805
      END                                                               CHA0806
      SUBROUTINE BDOKI(L,X,V,DV,MIN,NV)                        BDOK1    CHA0807
      IMPLICIT REAL*8(A-H,O-Z,$)                                        CHA0808
      DIMENSION X(L),V(L),DV(L),MIN(L)                                  CHA0809
      COMMON /AB/A(50)                                                  CHA0810
      EQUIVALENCE (LL,A(2)),(KEYMON,A(11))                              CHA0811
      COMMON /DRAW/GODELX,GODELY,UMIN,UMAX,PMIN,PMAX,ROMIN,ROMAX        CHA0812
     1             ,XMIN,XMAX,SMIN,SMAX,IVERSA                          CHA0813
      COMMON /MONIT/CASEAV(4),NC14(4),NF16(6),                          CHA0814
     1                 NMONU(4),NMONP(4),NMONRO(4),NMONZ(4)             CHA0815
      DIMENSION NMONV(4,4)                                              CHA0816
      EQUIVALENCE (NMONV(1,1),NMONU(1))                                 CHA0817
      DIMENSION NAMEV(4)                                                CHA0818
      DATA NAMEV/'U','P','RO','Z'/                                      CHA0819
      DATA EPS/1.D-9/                                                   CHA0820
C***********************************************************************CHA0821
      GO TO (1,2,3,4), NV                                               CHA0822
 1    AMIDA=(UMAX-UMIN)**2                                              CHA0823
      GO TO 9                                                           CHA0824
 2    AMIDA=(PMAX-PMIN)**2                                              CHA0825
      GO TO 9                                                           CHA0826
 3    AMIDA=(ROMAX-ROMIN)**2                                            CHA0827
      GO TO 9                                                           CHA0828
 4    AMIDA=1.D0                                                        CHA0829
      GO TO 9                                                           CHA0830
 9    CONTINUE                                                          CHA0831
      AMIDA=AMIDA*EPS**2                                                CHA0832
      EPSA=DSQRT(AMIDA)                                                 CHA0833
      DO 29 I=2,LL                                                      CHA0834
      ICAT=0                                                            CHA0835
      IF(DABS(DV(I)).LE.EPSA) DV(I)=0.                                  CHA0836
      IF(DV(I).EQ.0.) GO TO 29                                          CHA0837
      VLEFT=V(I)-0.5D0*DV(I)                                            CHA0838
      VRIGHT=V(I)+0.5D0*DV(I)                                           CHA0839
      VM=V(I-1)                                                         CHA0840
      VP=V(I+1)                                                         CHA0841
      SIGN=(VP-V(I))*(V(I)-VM)                                          CHA0842
      IF(SIGN.GT.-AMIDA) GO TO 22                                       CHA0843
  21  DV(I)=0.                                                          CHA0844
      ICAT=1                                                            CHA0845
      GO TO 20                                                          CHA0846
  22  CONTINUE                                                          CHA0847
      SIGN=(VP-VM)*DV(I)                                                CHA0848
      IF(SIGN.GT.-AMIDA) GO TO 24                                       CHA0849
  23  DV(I)=0.5D0*(VP-VM)                                               CHA0850
      VLEFT=V(I)-0.5D0*DV(I)                                            CHA0851
      VRIGHT=V(I)+0.5D0*DV(I)                                           CHA0852
      ICAT=2                                                            CHA0853
  24  SIGN=(VLEFT-VM)*DV(I)                                             CHA0854
      IF(SIGN.GT.-AMIDA) GO TO 26                                       CHA0855
  25  VLEFT=VM                                                          CHA0856
      VRIGHT=2.D0*V(I)-VLEFT                                            CHA0857
      DV(I)=VRIGHT-VLEFT                                                CHA0858
      ICAT=3                                                            CHA0859
  26  SIGN=(VP-VRIGHT)*DV(I)                                            CHA0860
      IF (SIGN.GT.-AMIDA) GO TO 28                                      CHA0861
  27  VRIGHT=VP                                                         CHA0862
      VLEFT=2.D0*V(I)-VRIGHT                                            CHA0863
      DV(I)=VRIGHT-VLEFT                                                CHA0864
      ICAT=3                                                            CHA0865
 28   IF(DABS(DV(I)).LE.0.5D0*DABS(VP-VM)) GO TO 31                     CHA0866
 30   DV(I)=0.5D0*(VP-VM)                                               CHA0867
      ICAT=4                                                            CHA0868
 31   CONTINUE                                                          CHA0869
 20   CONTINUE                                                          CHA0870
      IF (DABS(DV(I)).GT.EPSA) GO TO 40                                 CHA0871
      DV(I)=0.                                                          CHA0872
 40   CONTINUE                                                          CHA0873
      IF (ICAT.GT.0) NMONV(ICAT,NV)=NMONV(ICAT,NV)+1                    CHA0874
 29   CONTINUE                                                          CHA0877
```

```
      RETURN
      END
```

DCOLE

```
      SUBROUTINE DCOLE(L,X,V,DV,MIN,NV)
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION X(L),V(L),DV(L),MIN(L)
      COMMON /AB/A(50)
      EQUIVALENCE (LL,A(2))
C*********************************************************************
      DO 1 I=2,LL
      IM=I-1
      IP=I+1
      DV(I)=0.5D0*(V(IP)-V(IM))
    1 CONTINUE
      RETURN
      END
```

PRINT

```
      SUBROUTINE PRINT
     1               (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,
     2                US,PS,UIDOT,PIDOT,
     *               FIMZ,ZMDOT,
     3                TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION X(L),U(L),P(L),RO(L),G(L),E(L),DU(L),DP(L),DRO(L),
     1          DG(L),DXSI(L),MIN(L),
     2          US(L),PS(L),UIDOT(L),PIDOT(L)
     3         ,TENA(L),FIRO(L),FIM(L),FIE(L)
     4         ,GIP(L),VOL(L),Z(L),DZ(L)
     5         ,FIMZ(L),ZMDOT(L)
      COMMON /TOT/AMTOT,ETOT,EKTOT,EPTOT,TENTOT
      COMMON /STEP0/UL,PL,ROL,GL,UR,PR,ROR,GR,USTAR,PSTAR,
     1              RSTARL,RSTARR,GSTARL,GSTARR,
     2              CL,CR,CSTARL,CSTARR,SL,SR,WL,WR,UW(6)
     3             ,LAMDAL,LAMDAR,RATEL,RATER,TEMPL,TEMPR,TEMPSL,TEMPSR
     4             ,ZL,ZR,ZSTARL,ZSTARR,NFLUX,HELEML,HELEMR
      REAL*8 LAMDAL,LAMDAR
      LOGICAL HELEML,HELEMR
      COMMON /AB/A(50)
      EQUIVALENCE (LL,A(2)),(T,A(3)),(NCYC,A(12)),(DT,A(4))
      EQUIVALENCE (UGAL,A(15))
C     COMMON/DIFFUS/U2,P2,RO2,ARW
      COMMON/DETO/QDET,PCJDET,RCJDET,UCJDET,DCJDET,PODET,ROODET,
     1           RATE,TEMPC
      COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11
     1       ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23
     2       ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35
      REAL*8 NG,MU2
      COMMON /MONIT/CASEAV(4),NC14(4),NF16(6),
     1                NMONU(4),NMONP(4),NMONRO(4),NMONZ(4)
      DIMENSION CASAV1(4)
      LOGICAL FULLPR
C*********************************************************************
      FULLPR=.TRUE.
      PRINT 1
    1 FORMAT(1H1)
      PRINT 2, T,DT,NCYC
    2 FORMAT(1X,10X,'RESULTS AT  T=',D11.5,5X,'DT=',D11.5,5X,'NCYC=',
     1             I5//)
      PRINT 3, AMTOT,ETOT,EKTOT,EPTOT,TENTOT
    3 FORMAT(1X,'AMTOT=',D20.14,2X,'ETOT,EKTOT,EPTOT=',3D22.14/
     1     1X,'TENTOT=',D21.14//)
    4 FORMAT(1X,'  I','     X        ','      U       ','      P       ',
     1              '     RO       ','      G       ','      Z       ',
     2              '     DU       ','    DP      ','   DRO      ',
     3              '     DG       ','    DZ')
   44 FORMAT(1X,'    ','              ','     US       ','      PS      ',
     1              '    ZMDOT     ','     FIMZ     ','   AMDOT      ',
     2              '    AMDOTN    ','   TEMP     ','  ENTALP    ',
     3              '    AMACH     ','  ENTRO     ')
    5 FORMAT(1X)
      IF (UGAL.NE.0.) PRINT 6, UGAL
    6 FORMAT(/11X,'INITIAL VELOCITY CORRESPONDS TO  UGAL=',D15.6/)
      DO 10 I=1,L
      IF (MOD(I,10).NE.1) GO TO 11
```

```
      PRINT 5                                                           CHA0950
      PRINT 4                                                           CHA0951
      PRINT 44                                                          CHA0952
      PRINT 5                                                           CHA0953
 11   CONTINUE                                                          CHA0954
      PRINT 12,I,X(I),U(I),P(I),RO(I),G(I),Z(I),DU(I),DP(I),DRO(I),     CHA0955
     1          DG(I),DZ(I)                                             CHA0956
 12   FORMAT(1X,I3,6D12.5,5D11.4)                                       CHA0957
      ENTRO=P(I)/RO(I)**GAMA                                            CHA0958
      IF(.NOT.FULLPR) GO TO 131                                         CHA0959
      IF(I.EQ.1) GO TO 131                                              CHA0960
      IM=I-1                                                            CHA0961
      UL=U(IM)+0.5*DU(IM)                                               CHA0962
      PL=P(IM)+0.5*DP(IM)                                               CHA0963
      ROL=RO(IM)+0.5*DRO(IM)                                            CHA0964
      GL=G(IM)+0.5*DG(IM)                                               CHA0965
      CL=GL/ROL                                                         CHA0966
      ZL=Z(IM)+0.5*DZ(IM)                                               CHA0967
      IF(ZL.LT.0.) ZL=0.                                                CHA0968
      UR=U(I)-0.5*DU(I)                                                 CHA0969
      PR=P(I)-0.5*DP(I)                                                 CHA0970
      GR=G(I)-0.5*DG(I)                                                 CHA0971
      ROR=RO(I)-0.5*DRO(I)                                              CHA0972
      CR=GR/ROR                                                         CHA0973
      ZR=Z(I)-0.5*DZ(I)                                                 CHA0974
      IF(ZR.LT.0.) ZR=0.                                                CHA0975
      IF(PL.LE.0.) PL=1.D-8                                             CHA0976
      IF(PR.LE.0.) PR=1.D-8                                             CHA0977
      CALL RIEMAN(L,I,MIN)                                              CHA0978
      XI=X(I)                                                           CHA0987
      RSTAR=RSTARL                                                      CHA0988
      IF(USTAR.LT.0.) RSTAR=RSTARR                                      CHA0989
      ZSTAR=ZL                                                          CHA0990
      IF(USTAR.LT.0.) ZSTAR=ZR                                          CHA0991
      AMACH=USTAR/DSQRT(GAMA*PSTAR/RSTAR)                               CHA0992
      AMDOT=RSTAR*USTAR*CROSS(XI)                                       CHA0993
      IF(I.NE.2) GO TO 132                                              CHA0994
      AMDOT0=AMDOT                                                      CHA0995
      IF(DABS(AMDOT0).LT.1.D-12) AMDOT0=1.D0                            CHA0996
 132  CONTINUE                                                          CHA0997
      AMDOTN=AMDOT/AMDOT0                                               CHA0998
      ENTALP=(GAMA/(GAMA-1.D0))*PSTAR/RSTAR+0.5D0*USTAR**2+QDET*ZSTAR   CHA0999
      ARW=1.D0                                                          CHA1000
      TEMP=PSTAR/(RSTAR*ARW)                                            CHA1001
      PRINT 13,US(I),PS(I),                                             CHA1002
     1          ZMDOT(I),FIMZ(I),AMDOT,AMDOTN,TEMP,ENTALP,AMACH,ENTRO   CHA1003
 13   FORMAT(4X,12X,5D12.5,6D11.4)                                      CHA1004
 131  CONTINUE                                                          CHA1005
 10   CONTINUE                                                          CHA1006
C  JOB STATISTICS                                                       CHA1007
      DO 40 I=1,4                                                       CHA1008
      CASAV1(I)=0.                                                      CHA1009
      IF (NC14(I).NE.0) CASAV1(I)=CASEAV(I)/DFLOAT(NC14(I))             CHA1010
 40   CONTINUE                                                          CHA1011
      PRINT 30                                                          CHA1012
 30   FORMAT(///1X,10('*'),3X,'JOB STATISTICS',3X,10('*')//)            CHA1013
      PRINT 31,(NC14(I),I=1,4)                                          CHA1014
 31   FORMAT(1X,'NO. OF VARIOUS CASES IN RIEMAN SOLVER   NC14(NCASE)=', CHA1015
     1          4I10)                                                   CHA1016
      PRINT 301, (CASAV1(I),I=1,4)                                      CHA1017
 301  FORMAT(/1X,'AVERAGE NUMBER OF ITERATIONS IN RIEMAN SOLVER',       CHA1018
     1          1X,'  CASAV1(NCASE)=',4(F6.2,4X))                       CHA1019
      PRINT 32,(NF16(I),I=1,6)                                          CHA1020
 32   FORMAT(/1X,'NO. OF VARIOUS FLUX CASES   NF16(NFLUX)=',6I10)       CHA1021
      ICAT0=4                                                           CHA1022
      PRINT 33,(NMONU(I),I=1,ICAT0),(NMONP(I),I=1,ICAT0),               CHA1023
     1          (NMONRO(I),I=1,ICAT0),(NMONZ(I),I=1,ICAT0)              CHA1024
 33   FORMAT(/1X,'NO. OF MONOTONICITY INTERVENTIONS FOR EACH VAR.',     CHA1025
     1          1X,'IN EACH CATEGORY.'/                                 CHA1026
     1          1X,'NMONU (ICAT)=',4I10/                                CHA1027
     1          1X,'NMONP (ICAT)=',4I10/                                CHA1028
     1          1X,'NMONRO(ICAT)=',4I10/                                CHA1029
```

```
     1          1X,'NMONZ (ICAT)=',4I10/)                                CHA1030
      RETURN                                                             CHA1031
      END                                                                CHA1032
      SUBROUTINE SOF(ISTOP)                                              CHA1034
      IMPLICIT REAL*8(A-H,O-Z,$)                                         CHA1035
      DIMENSION ISTOP(1)                                                 CHA1036
      PRINT 1, ISTOP                                                     CHA1037
    1 FORMAT(//1X,3H***,3X,20A4,3X,3H***///)                             CHA1038
      PRINT 1                                                            CHA1039
      XX=-1.D0                                                           CHA1040
      YY=DSQRT(XX)                                                       CHA1041
      STOP                                                               CHA1042
      END                                                                CHA1043
      SUBROUTINE RIEMAN(L,I,MIN)                                         CHA1310
      IMPLICIT REAL*8(A-H,O-Z,$)                                         CHA1311
      DIMENSION MIN(L)                                                   CHA1312
      COMMON /STEP0/UL,PL,ROL,GL,UR,PR,ROR,GR,USTAR,PSTAR,               CHA1313
     1              RSTARL,RSTARR,GSTARL,GSTARR,                         CHA1314
     2              CL,CR,CSTARL,CSTARR,SL,SR,WL,WR,UW(6)                CHA1315
     3             ,LAMDAL,LAMDAR,RATEL,RATER,TEMPL,TEMPR,TEMPSL,TEMPSR  CHA1316
     4             ,ZL,ZR,ZSTARL,ZSTARR,NFLUX,HELEML,HELEMR              CHA1317
      REAL*8 LAMDAL,LAMDAR                                               CHA1318
      LOGICAL HELEML,HELEMR                                              CHA1319
      COMMON /STEP1/DUIDT,DPIDT,DGIDTL,DGIDTR,DRIDTL,DRIDTR              CHA1320
     2  ,ASTARL,ASTARR,LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR             CHA1321
     3   ,RAT,SH                                                         CHA1322
     4   ,BETACL,BETACR,DSDASL,DSDASR,DZDASL,DZDASR                      CHA1323
      REAL*8 LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR                       CHA1324
      COMMON /DRAW/GODELX,GODELY,UMIN,UMAX,PMIN,PMAX,ROMIN,ROMAX         CHA1325
     1             ,XMIN,XMAX,SMIN,SMAX,IVERSA                           CHA1326
      COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11         CHA1327
     1        ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23           CHA1328
     2        ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35           CHA1329
      REAL*8 NG,MU2                                                      CHA1330
      COMMON /AB/A(50)                                                   CHA1331
      COMMON /MONIT/CASEAV(4),NC14(4),NF16(6),                           CHA1332
     1                  NMONU(4),NMONP(4),NMONRO(4),NMONZ(4)             CHA1333
C***********************************************************************CHA1334
      DATA NMAX/63/                                                      CHA1335
      DATA EPS/1.D-8/                                                    CHA1336
      DATA NTRY/0/                                                       CHA1337
C***********************************************************************CHA1338
      UW(6)=1.D20                                                        CHA1339
      WL=0.                                                              CHA1340
      WR=0.                                                              CHA1341
      ZETAL=PL**G8                                                       CHA1342
      ZETAR=PR**G8                                                       CHA1343
      CLG=CL/GAMA                                                        CHA1344
      CRG=CR/GAMA                                                        CHA1345
      ZSTARL=ZL                                                          CHA1346
      ZSTARR=ZR                                                          CHA1347
      IF (ZETAL.LT.ZETAR) GO TO 102                                      CHA1348
C  LEFT PRESSURE IS HIGHER                                               CHA1349
  101 CONTINUE                                                           CHA1350
      EVERR=(PL-PR)/PR                                                   CHA1351
      USR=UR+CRG*EVERR/DSQRT(1.D0+G6*EVERR)                              CHA1352
      SRR=USR                                                            CHA1353
      UEL=UL-G7*CL*(ZETAR-ZETAL)/ZETAL                                   CHA1354
      SLL=UEL                                                            CHA1355
      NL=2                                                               CHA1356
      NR=2                                                               CHA1357
      IF (USR.GE.UL) NL=1                                                CHA1358
      IF (UEL.LE.UR) NR=1                                                CHA1359
      IF (DABS(EVERR).LT.EPS) GO TO 100                                  CHA1360
      IF (NL.EQ.2.AND.NR.EQ.1) GO TO 7001                                CHA1361
      GO TO 100                                                          CHA1362
C  RIGHT PRESSURE IS HIGHER                                              CHA1363
  102 CONTINUE                                                           CHA1364
      EVERL=(PR-PL)/PL                                                   CHA1365
      USL=UL-CLG*EVERL/DSQRT(1.D0+G6*EVERL)                              CHA1366
      SLL=USL                                                            CHA1367
      UER=UR+G7*CR*(ZETAL-ZETAR)/ZETAR                                   CHA1368
```

SOF

RIEMAN

```
      SRR=UER
      NL=2
      NR=2
      IF (UER.GE.UL) NL=1
      IF (USL.LE.UR) NR=1
      IF (DABS(EVERL).LT.EPS) GO TO 100
      IF (NL.EQ.1.AND.NR.EQ.2) GO TO 7001
      GO TO 100
 100  CONTINUE
      IF (NL.EQ.1.AND.NR.EQ.2) NCASE=1
      IF (NL.EQ.2.AND.NR.EQ.2) NCASE=2
      IF (NL.EQ.2.AND.NR.EQ.1) NCASE=3
      IF (NL.EQ.1.AND.NR.EQ.1) NCASE=4
      IF(DABS(PL-PR)+DABS(UL-UR).LT.EPS*(PMAX-UMIN)) NCASE=4
      UMIDA=EPS*DMAX1(CL,CR)
      DUDZL=-G7*CL/ZETAL
      DUDZR= G7*CR/ZETAR
      ZETA=(-(UR-UL)+ZETAR*DUDZR-ZETAL*DUDZL)/(DUDZR-DUDZL)
      IF (ZETA.LE.0.) GO TO 7002
      N=0
      GO TO (1,2,3,4), NCASE
C  THE CASE  ES
 1    ITYPE=NCASE
      HELEML=.FALSE.
      HELEMR=.TRUE.
 11   N=N+1
      IF (N.GT.NMAX) GO TO 7003
      ZETAF=ZETA
      UEL=UL-G7*CL*(ZETAF-ZETAL)/ZETAL
      PPR=(ZETAF/ZETAR)**NG
      EVERR=PPR-1.D0
      SQRR=DSQRT(1.D0+G6*EVERR)
      USR=UR+CRG*EVERR/SQRR
      DU=UEL-USR
      IF (DABS(DU).LE.UMIDA) GO TO 10
      DUDZR=NG*CRG*(PPR/ZETAF)*(1.D0+G9*EVERR)/SQRR**3
      ZETA=ZETAF+DU/(DUDZR-DUDZL)
      GO TO 11
 10   CONTINUE
      USTAR=(UEL+USR)/2.D0
      IF(DABS(USTAR).LT.EPS*UMAX) USTAR=0.
      PSTAR=PPR*PR
      CSTARL=CL+(UL-USTAR)/G7
      RSTARL=GAMA*PSTAR/CSTARL**2
      GSTARL=CSTARL*RSTARL
C  EQU. NO.  69.01 OF THE BOOK BY COURANT-FRIEDRICHS.
      WWR=G11*(USTAR-UR)*ROR
      WR=WWR+DSQRT(GR**2+WWR**2)
      RSTARR=ROR*WR/(WR-ROR*(USTAR-UR))
      GSTARR=DSQRT(GAMA*PSTAR*RSTARR)
      CSTARR=GSTARR/RSTARR
      WRE=WR/ROR+UR
      UW(1)=UL-CL
      UW(2)=USTAR-CSTARL
      UW(3)=USTAR
      UW(4)=WRE
      UW(5)=WRE
      GO TO 5
C  THE CASE  SS
 2    ITYPE=NCASE
      HELEML=.TRUE.
      HELEMR=.TRUE.
 21   N=N+1
      IF (N.GT.NMAX) GO TO 7003
      ZETAF=ZETA
      PF=ZETAF**NG
      PPL=PF/PL
      PPR=PF/PR
      EVERL=PPL-1.D0
      EVERR=PPR-1.D0
      SQRL=DSQRT(1.D0+G6*EVERL)
      SQRR=DSQRT(1.D0+G6*EVERR)
```

```
      USL=UL-CLG*EVERL/SQRL
      USR=UR+CRG*EVERR/SQRR
      DU=USL-USR
      IF (DABS(DU).LE.UMIDA) GO TO 20
      DUDZL=-NG*CLG*(PPL/ZETAF)*(1.D0+G9*EVERL)/SQRL**3
      DUDZR= NG*CRG*(PPR/ZETAF)*(1.D0+G9*EVERR)/SQRR**3
      ZETA=ZETAF+DU/(DUDZR-DUDZL)
      GO TO 21
 20   CONTINUE
      USTAR=(USL+USR)/2.D0
      IF(DABS(USTAR).LT.EPS*UMAX) USTAR=0.
      PSTAR=(PPL*PL+PPR*PR)/2.D0
      WWR=G11*(USTAR-UR)*ROR
      WR=WWR+DSQRT(GR**2+WWR**2)
      WWL=-G11*(USTAR-UL)*ROL
      WL=WWL+DSQRT(GL**2+WWL**2)
      RSTARL=ROL*WL/(WL+ROL*(USTAR-UL))
      RSTARR=ROR*WR/(WR-ROR*(USTAR-UR))
      GSTARL=DSQRT(GAMA*PSTAR*RSTARL)
      GSTARR=DSQRT(GAMA*PSTAR*RSTARR)
      CSTARL=GSTARL/RSTARL
      CSTARR=GSTARR/RSTARR
      WLE=-WL/ROL+UL
      WRE=WR/ROR+UR
      UW(1)=WLE
      UW(2)=WLE
      UW(3)=USTAR
      UW(4)=WRE
      UW(5)=WRE
      GO TO 5
C  THE CASE  SE
 3    ITYPE=NCASE
      HELEML=.TRUE.
      HELEMR=.FALSE.
 31   N=N+1
      IF (N.GT.NMAX) GO TO 7003
      ZETAF=ZETA
      UER=UR+G7*CR*(ZETAF-ZETAR)/ZETAR
      PPL=(ZETAF/ZETAL)**NG
      EVERL=PPL-1.D0
      SQRL=DSQRT(1.D0+G6*EVERL)
      USL=UL-CLG*EVERL/SQRL
      DU=USL-UER
      IF (DABS(DU).LE.UMIDA) GO TO 30
      DUDZL=-NG*CLG*(PPL/ZETAF)*(1.D0+G9*EVERL)/SQRL**3
      ZETA=ZETAF+DU/(DUDZR-DUDZL)
      GO TO 31
 30   CONTINUE
      USTAR=(USL+UER)/2.D0
      IF(DABS(USTAR).LT.EPS*UMAX) USTAR=0.
      PSTAR=PPL*PL
      CSTARR=CR-(UR-USTAR)/G7
      RSTARR=GAMA*PSTAR/CSTARR**2
      GSTARR=CSTARR*RSTARR
      WWL=-G11*(USTAR-UL)*ROL
      WL=WWL+DSQRT(GL**2+WWL**2)
      WLE=-WL/ROL+UL
      RSTARL=ROL*WL/(WL+ROL*(USTAR-UL))
      GSTARL=DSQRT(GAMA*PSTAR*RSTARL)
      CSTARL=GSTARL/RSTARL
      UW(1)=WLE
      UW(2)=WLE
      UW(3)=USTAR
      UW(4)=USTAR+CSTARR
      UW(5)=UR+CR
      GO TO 5
C  THE CASE  EE
 4    ITYPE=NCASE
      HELEML=.FALSE.
      HELEMR=.FALSE.
      PSTAR=ZETA**NG
      USTAR=UL-G7*CL*(ZETA-ZETAL)/ZETAL
```

```
      IF(DABS(USTAR).LT.EPS*UMAX) USTAR=0.                              CHA1513
      CSTARL=CL+(UL-USTAR)/G7                                           CHA1514
      CSTARR=CR-(UR-USTAR)/G7                                           CHA1515
      RSTARL=GAMA*PSTAR/CSTARL**2                                       CHA1516
      RSTARR=GAMA*PSTAR/CSTARR**2                                       CHA1517
      GSTARL=RSTARL*CSTARL                                              CHA1518
      GSTARR=RSTARR*CSTARR                                              CHA1519
      UW(1)=UL-CL                                                       CHA1520
      UW(2)=USTAR-CSTARL                                                CHA1521
      UW(3)=USTAR                                                       CHA1522
      UW(4)=USTAR+CSTARR                                                CHA1523
      UW(5)=UR+CR                                                       CHA1524
      N=1                                                               CHA1525
      GO TO 5                                                           CHA1526
    5 CONTINUE                                                          CHA1527
      DO 6 K=1,6                                                        CHA1528
      NFLUX=K                                                           CHA1529
      IF (UW(K).GE.0.) GO TO 61                                         CHA1530
    6 CONTINUE                                                          CHA1531
      NFLUX=6                                                           CHA1532
   61 CONTINUE                                                          CHA1533
      NC14(NCASE)=NC14(NCASE)+1                                         CHA1537
      CASEAV(NCASE)=CASEAV(NCASE)+DFLOAT(N)                             CHA1538
      NF16(NFLUX)=NF16(NFLUX)+1                                         CHA1539
      IF(NTRY.GE.2)GO TO 666                                            CHA1540
      IF(I.NE.2.AND.I.NE.L) GO TO 666                                   CHA1541
      PRINT 667,I,NFLUX,NCASE,PL,UL,ROL,PR,UR,ROR,USTAR,PSTAR,RSTARL,   CHA1542
     1          RSTARR,(KK,UW(KK),KK=1,6)                               CHA1543
  667 FORMAT(/1X,'I,NFLUX,NCASE=',3I5/1X,'PL,UL,ROL,PR,UR,ROR=',6D12.4/ CHA1544
     1        1X,'USTAR,PSTAR,RSTARL,RSTARR=',4D13.4/                   CHA1545
     2        1X,'KK,UW(KK)=',6(I4,2X,D13.4)/)                          CHA1546
      NTRY=NTRY+1                                                       CHA1547
  666 CONTINUE                                                          CHA1548
      RETURN                                                            CHA1549
 7001 CONTINUE                                                          CHA1550
      PRINT 7101, PL,UL,PR,UR,ZETAL,ZETAR,SLL,SRR,NL,NR,I               CHA1551
 7101 FORMAT(//1X,'FROM RIEMAN.  AN IMPOSSIBLE CASE OF EXPANSION/SHOCK' CHA1552
     1        //1X,'PL,UL,PR,UR=',4D25.14//                             CHA1553
     2          1X,'ZETAL,ZETAR,SLL,SRR=',4D25.14//                     CHA1554
     3          1X,'NL,NR,I=',3I10//)                                   CHA1555
      CALL SOF('7001')                                                  CHA1556
 7002 CONTINUE                                                          CHA1557
      PRINT 7102, ZETA,DUDZL,DUDZR,ZETAL,ZETAR,PL,UL,PR,UR,N,NCASE,I    CHA1558
 7102 FORMAT(//1X,'FROM RIEMAN.  NEGATIVE PRESSURE AT THE INTERSECTION',CHA1559
     1          1X,'OF L AND R EXPANSION BRANCHES'//                    CHA1560
     2          1X,'IT MEANS THAT A CAVITATION TENDS TO FORM.  THIS',   CHA1561
     3          1X,'POSSIBILITY IS EXCLUDED IN PRESENT VERSION'//       CHA1562
     4          1X,'ZETA,DUDZL,DUDZR,ZETAL,ZETAR,PL,UL,PR,UR=',9D10.3// CHA1563
     5          1X,'N,NCASE,I=',3I10//)                                 CHA1564
      CALL SOF('7002')                                                  CHA1565
 7003 CONTINUE                                                          CHA1566
      PRINT 7103, I,N,NCASE,DU,UMIDA,EPS,PL,UL,PR,UR,                   CHA1567
     1            ZETA,ZETAF,ZETAL,ZETAR,DUDZL,DUDZR                    CHA1568
 7103 FORMAT(//1X,'FROM RIEMAN.  NUMBER OF ITERATIONS EXCEEDED.'//      CHA1569
     1          1X,'I,N,NCASE,DU,UMIDA,EPS=',3I6,3D18.6//               CHA1570
     2          1X,'PL,UL,PR,UR,ZETA,ZETAF=',6D18.10//                  CHA1571
     3          1X,'ZETAL,ZETAR,DUDZL,DUDZR=',4D18.10//)                CHA1572
      CALL SOF('7003')                                                  CHA1573
      RETURN                                                            CHA1574
      END                                                               CHA1575
C$OPTIONS LIST                                                           CHA1576
      SUBROUTINE MAGA(L,I,MIN)                          MAGA            CHA1577
      IMPLICIT REAL*8(A-H,O-Z,$)                                        CHA1578
      DIMENSION MIN(L)                                                  CHA1579
      COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11        CHA1580
     1          ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23        CHA1581
     2          ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35        CHA1582
      REAL*8 NG,MU2                                                     CHA1583
      COMMON/DETO/QDET,PCJDET,RCJDET,UCJDET,DCJDET,PODET,ROODET,        CHA1584
     1            RATE,TEMPC                                            CHA1585
      COMMON /STEPO/UL,PL,ROL,GL,UR,PR,ROR,GR,USTAR,PSTAR,              CHA1586
     1              RSTARL,RSTARR,GSTARL,GSTARR,                        CHA1587
```

```
     2              CL,CR,CSTARL,CSTARR,SL,SR,WL,WR,UW(6)
     3             ,LAMDAL,LAMDAR,RATEL,RATER,TEMPL,TEMPR,TEMPSL,TEMPSR
     4             ,ZL,ZR,ZSTARL,ZSTARR,NFLUX,HELEML,HELEMR
      REAL*8 LAMDAL,LAMDAR
      LOGICAL HELEML,HELEMR
      COMMON /STEP1/DUIDT,DPIDT,DGIDTL,DGIDTR,DRIDTL,DRIDTR
     2 ,ASTARL,ASTARR,LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR
     3  ,RAT,SH
     4  ,BETACL,BETACR,DSDASL,DSDASR,DZDASL,DZDASR
      REAL*8 LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR
      COMMON /GRADS/DUDXIL,DPDXIL,DGDXIL,DRDXIL,DZDXIL,DSDXIL,
     1              DUDXIR,DPDXIR,DGDXIR,DRDXIR,DZDXIR,DSDXIR
      COMMON /AB/A(50)
      REAL*8 LU,LP,LRO,LLAMDA
      DATA EPS/1.D-6/
C*************************************************************************
C  WE HERE SOLVE FOR THE TIME-DERIVATIVES ALONG THE CONTACT SURFACE,
C  NAMELY  DUIDT,DPIDT.  FROM THESE WE ALSO OBTAIN THE OTHER
C  TIME-DERIVATIVES (SEE COMMON /STEP1/).
C  WE COMPUTE THE COEFFICIENTS FOR TWO EQUATIONS FOR DUIDT,DPIDT.  THESE
C  ARE          AAL*DUIDT+BBL*DPIDT=DDL
C               AAR*DUIDT+BBR*DPIDT=DDR
C*************************************************************************
      IF(SH.LE.EPS)RAT=0.
C
C  LEFT SIDE OF CONTACT
C
      IF (.NOT.HELEML) GO TO 12
 11   CONTINUE
C  LEFT SHOCK
      DP=PSTAR-PL
      DU=USTAR-UL
      Z2=0.5D0/(PSTAR+MU2*PL)
      LU=DU*(0.5D0*ROL+MU2*Z2*GL**2)-GL**2/WL-WL
      LRO=-0.5D0*DP/ROL
      LP=-2.D0-MU2*Z2*DP
      AAL=2.D0-Z2*DP
      BBL=Z2*DU+WL/GSTARL**2+1.D0/WL
      DDL=LU*DUDXIL+LRO*DRDXIL+LP*DPDXIL
      DDL=DDL-WL*USTAR*RAT/RSTARL
     1  +UL*RAT*(-GAMA*PL/WL+DU*(GAMA*PL*MU2*Z2+0.5D0))
      GO TO 10
 12   CONTINUE
C  LEFT RAREFACTION
      A1=DUDXIL+DPDXIL/GL
      BETA=GSTARL/CL
      SQB=DSQRT(BETA)
      ASTARL=A1-(CL/(G15*SL))*DSDXIL*(BETA**G5-1.D0)
      AAL=1.D0
      BBL=1.D0/GSTARL
      DDL=-GSTARL*ASTARL/SQB
      DSDAL=DSDXIL
      DZDAL=DZDXIL
      DSDASL=DSDXIL*SQB
      DZDASL=DZDXIL*SQB
      GEOM=RAT*((GAMA-1.D0)*UL+2.D0*CL)*
     1 (BETA**G13-1.D0)/(ROL*(GAMA-3.D0))
     1  -4.D0*RAT*CL*(BETA**G14-1.D0)/(ROL*(3.D0*GAMA-5.D0))
      ASTARL=ASTARL-GEOM
      EVER1= GSTARL*GEOM/SQB
      EVER2=-RAT*USTAR*CSTARL
      DDL=DDL+EVER1+EVER2
      GO TO 10
 10   CONTINUE
C
C  RIGHT SIDE OF CONTACT
C
      IF (.NOT.HELEMR) GO TO 22
 21   CONTINUE
C  RIGHT SHOCK
      DP=PSTAR-PR
      DU=USTAR-UR
```

```
      Z2=0.5D0/(PSTAR+MU2*PR)
      LU=DU*(0.5D0*ROR+MU2*Z2*GR**2)+GR**2/WR+WR
      LRO=-0.5D0*DP/ROR
      LP=-2.D0-MU2*Z2*DP
      AAR=2.D0-Z2*DP
      BBR=Z2*DU-WR/GSTARR**2-1.D0/WR
      DDR=LU*DUDXIR+LRO*DRDXIR+LP*DPDXIR
      DDR=DDR+WR*USTAR*RAT/RSTARR
     1    +UR*RAT*(GAMA*PR/WR+DU*(GAMA*PR*MU2*Z2+0.5D0))
      GO TO 20
   22 CONTINUE
C  RIGHT RAREFACTION
      A1=DUDXIR-DPDXIR/GR
      BETA=GSTARR/GR
      SQB=DSQRT(BETA)
      ASTARR=A1+(CR/(G15*SR))*DSDXIR*(BETA**G5-1.D0)
      AAR=1.D0
      BBR=-1.D0/GSTARR
      DDR=GSTARR*ASTARR/SQB
      DSDAR=DSDXIR
      DZDAR=DZDXIR
      DSDASR=DSDXIR*SQB
      DZDASR=DZDXIR*SQB
      GEOM=RAT*(-(GAMA-1.D0)*UR+2.D0*CR)*(BETA**G13-1.D0)
     1  /(ROR*(GAMA-3.D0))
     2  -4.D0*RAT*CR*(BETA**G14-1.D0)/(ROR*(3.D0*GAMA-5.D0))
      ASTARR=ASTARR+GEOM
      EVER1=GSTARR*GEOM/SQB
      EVER2=RAT*USTAR*CSTARR
      DDR=DDR+EVER1+EVER2
      GO TO 20
   20 CONTINUE
      DET=AAL*BBR-AAR*BBL
      DUIDT=(DDL*BBR-DDR*BBL)/DET
      DPIDT=-(DDL*AAR-DDR*AAL)/DET
      DRIDTL=DPIDT/CSTARL**2
      DRIDTR=DPIDT/CSTARR**2
      RETURN
      END
      SUBROUTINE FLUXE(L,I,MIN)                        FLUXE
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION MIN(L)
      COMMON /AB/A(50)
      EQUIVALENCE (DT,A(4)),(NCYC,A(12))
      COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11
     1           ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23
     2           ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35
      REAL*8 NG,MU2
      COMMON /GRADS/ DUDXIL,DPDXIL,DGDXIL,DRDXIL,DZDXIL,DSDXIL,
     1               DUDXIR,DPDXIR,DGDXIR,DRDXIR,DZDXIR,DSDXIR
      COMMON /STEP0/UL,PL,ROL,GL,UR,PR,ROR,GR,USTAR,PSTAR,
     1              RSTARL,RSTARR,GSTARL,GSTARR,
     2              CL,CR,CSTARL,CSTARR,SL,SR,WL,WR,UW(6)
     3             ,LAMDAL,LAMDAR,RATEL,RATER,TEMPL,TEMPR,TEMPSL,TEMPSR
     4             ,ZL,ZR,ZSTARL,ZSTARR,NFLUX,HELEML,HELEMR
      REAL*8 LAMDAL,LAMDAR
      LOGICAL HELEML,HELEMR
      COMMON /STEP1/DUIDT,DPIDT,DGIDTL,DGIDTR,DRIDTL,DRIDTR
     2  ,ASTARL,ASTARR,LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR
     3   ,RAT,SH
     4   ,BETACL,BETACR,DSDASL,DSDASR,DZDASL,DZDASR
      REAL*8 LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR
      COMMON/DETO/QDET,PCJDET,RCJDET,UCJDET,CCJDET,PODET,ROODET,
     1            RATE,TEMPC
      COMMON /FI/FIH1,FIH2,FIH3,UXN,PXN,GXN,ROXN,ZXN
     1   ,GIH
     2   ,FIH4,ZMDOTL,ZMDOTR
      REAL*8 LAMDA0
C***********************************************************************
C  RO,U,P,Z AND THEIR (XI,T) DERIVATIVES AT EULERIAN POINT X=X(I).
C***********************************************************************
      DT2=DT/2.D0
```

```
      GO TO (1,2,3,4,5,6),NFLUX
    1 CONTINUE
C
C  NFLUX=1.   LINE X=0 IS TO THE LEFT OF LEFT WAVE.
C
      UX=UL
      PX=PL
      ROX=ROL
      ZX=ZL
      GX=GL
      DUDXIX=DUDXIL
      DPDXIX=DPDXIL
      DRDXIX=DRDXIL
      DZDXIX=DZDXIL
      DUDTX=-DPDXIL
      DRODTX=-ROL**2*DUDXIL
      DPDTX=-GL**2*DUDXIL
      DRODTX=DRODTX-RAT*ROL*UL
      DPDTX=DRODTX*CL**2
      DZDTX=0.
      GO TO 9
    6 CONTINUE
C
C  NFLUX=6.   LINE X=0 IS TO THE RIGHT OF RIGHT WAVE.
C
      UX=UR
      PX=PR
      ROX=ROR
      ZX=ZR
      GX=GR
      DUDXIX=DUDXIR
      DPDXIX=DPDXIR
      DRDXIX=DRDXIR
      DZDXIX=DZDXIR
      DUDTX=-DPDXIR
      DPDTX=-GR**2*DUDXIR
      DRODTX=-ROR**2*DUDXIR
      DRODTX=DRODTX-RAT*ROR*UR
      DPDTX=DRODTX*CR**2
      DZDTX=0.
      GO TO 9
    2 CONTINUE
C
C  NFLUX=2.   SONIC CASE (LEFT).
C
      BETA0=(MU2*(UL/CL+G7))**(1.D0/MU2)
      SQB0=DSQRT(BETA0)
      A1=DUDXIL+DPDXIL/GL
      A0=A1-(CL/(G15*SL))*DSDXIL*(BETA0**G5-1.D0)
      EVER1=-((GAMA-1.D0)*UL+2.D0*CL)*(BETA0**G13-1.D0)/(GAMA-3.D0)
      EVER2=4.D0*CL*(BETA0**G14-1.D0)/(3.D0*GAMA-5.D0)
      EVER=(EVER1+EVER2)*RAT/ROL
      A0=(A0+EVER)
      DPDAX=GL*BETA0*A0
      C0=MU2*(UL+G7*CL)
      IF(C0.LT.0.) CALL SOF('FLUXE 2.  C0 NEGATIVE.')
      UX=C0
      ROX=GL*BETA0/C0
      ZX=ZL
      PX=ROX*C0**2/GAMA
      GX=ROX*C0
      DPDAX=DPDAX+RAT*UX*C0*SQB0
      DUDBX=-CL*BETA0**(-1.D0/G4)/G4
      DPDBX=PL*BETA0**MU2/G6
      DRODBX=ROL*BETA0**(-MU2)/G4
      DSDAX=SQB0*DSDAL
      DZDAX=SQB0*DZDAL
      DRODAX=DPDAX/C0**2-(ROX/(GAMA*SL))*DSDAX
      DUDAX=A0
      DGDAX=0.5D0*GAMA*(PX*DRODAX+ROX*DPDAX)/GX
      GO TO 9
    5 CONTINUE
```

```
C
C  NFLUX=5.   SONIC CASE (RIGHT).
C
      BETA0=(MU2*(-UR/CR+G7))**(1.D0/MU2)
      SQB0=DSQRT(BETA0)
      A1=DUDXIR-DPDXIR/GR
      A0=A1+(CR/(G15*SR))*DSDXIR*(BETA0**G5-1.D0)
      EVER1=(-(GAMA-1.D0)*UR+2.D0*CR)*(BETA0**G13-1.D0)/(GAMA-3.D0)
      EVER2=-4.D0*CR*(BETA0**G14-1.D0)/(3.D0*GAMA-5.D0)
      EVER=(EVER1+EVER2)*RAT/ROR
      A0=(A0+EVER)
      DPDAX=-GR*BETA0*A0
      C0=MU2*(-UR+G7*CR)
      IF(C0.LT.0.) CALL SOF('FLUXE 5.  C0 NEGATIVE.')
      UX=-C0
      ROX=GR*BETA0/C0
      ZX=ZR
      PX=ROX*C0**2/GAMA
      GX=ROX*C0
      DPDAX=DPDAX-RAT*UX*C0*DSQRT(BETA0)
      DUDBX=CR*BETA0**(-1.D0/G4)/G4
      DPDBX=PR*BETA0**MU2/G6
      DRODBX=ROR*BETA0**(-MU2)/G4
      DSDAX=SQB0*DSDAR
      DZDAX=SQB0*DZDAR
      DRODAX=DPDAX/C0**2-(ROX/(GAMA*SR))*DSDAX
      DUDAX=A0
      DGDAX=0.5D0*GAMA*(PX*DRODAX+ROX*DPDAX)/GX
      GO TO 9
  3   CONTINUE
C
C  NFLUX=3.   LINE X=0 IS BETWEEN THE LEFT WAVE AND THE CONTACT.
C
      UX=USTAR
      PX=PSTAR
      ROX=RSTARL
      ZX=ZL
      GX=GSTARL
      DUDXIX=-DPIDT/GSTARL**2
      DPDXIX=-DUIDT
      DUDXIX=DUDXIX-RAT*USTAR/RSTARL
      DZDXIX=DZDXIL
      DZDTX=0.
      IF (.NOT.HELEML) GO TO 32
 31   CONTINUE
C  LEFT SHOCK.
      DRDXIX=(RSTARL/WL)**2*(3.D0*DUIDT
     1          +DPIDT*(1.D0+3.D0*(WL/GSTARL)**2)/WL
     2          +DUDXIL*WL*((GL/WL)**2+3.D0)+3.D0*DPDXIL
     3          +DRDXIL*(WL/ROL)**2)
      EVER1=UL*RSTARL**2*RAT*((GL/WL)**2+1.D0)/(ROL*WL)
      EVER2=2.D0*RSTARL*USTAR*RAT/WL
      DRDXIX=DRDXIX+EVER1+EVER2
      DRODTX=-DUDXIX*ROX**2
      GO TO 33
 32   CONTINUE
      BETA=GSTARL/GL
      SQB=DSQRT(BETA)
      DPDA=ASTARL*GSTARL
      DPDA=GSTARL*(ASTARL+RAT*USTAR*CSTARL/(GL*   SQB))
      G41=1.D0/G4+0.5D0
      DRODA=(DRDXIL-DPDXIL/(CL*CL))   *BETA**G41+DPDA/(CSTARL**2)
      DRDXIX=   DRODA/SQB+DPIDT/(GSTARL*CSTARL**2)
      DRODA=DPDA/CSTARL**2-(RSTARL/(GAMA*SL))*DSDASL
      DRODTX=-DUDXIX*ROX**2
      DRDXIX=DRODA/SQB+DRODTX/GSTARL
 33   CONTINUE
      DUDTX=DUIDT
      DPDTX=DPIDT
      GO TO 9
  4   CONTINUE
C
```

```
C  NFLUX=6.   LINE X=0 IS BETWEEN THE CONTACT AND THE RIGHT WAVE.
C
      DPDXIX=-DUIDT
      UX=USTAR
      PX=PSTAR
      ROX=RSTARR
      ZX=ZR
      GX=GSTARR
      DUDXIX=-DPIDT/GSTARR**2
      DUDXIX=DUDXIX-RAT*USTAR/RSTARR
      DPDXIX=-DUIDT
      DZDXIX=DZDXIL
      DZDTX=0.
      IF (.NOT.HELEMR) GO TO 42
 41   CONTINUE
C  RIGHT SHOCK
      DRDXIX=(RSTARR/WR)**2*(3.*DUIDT
     1         -DPIDT*(1.D0+3.D0*(WR/GSTARR)**2)/WR
     2         -DUDXIR*WR*((GR/WR)**2+3.D0)+3.D0*DPDXIR
     3         +DRDXIR*(WR/ROR)**2)
      EVER1=UR*RSTARR**2*RAT*((GR/WR)**2+1.D0)/(ROR*WR)
      EVER2=2.D0*RSTARR*USTAR*RAT/WR
      DRDXIX=DRDXIX-EVER1-EVER2
      DRODTX=-DUDXIX*ROX**2
      GO TO 43
 42   CONTINUE
C  RIGHT RAREFACTION
      BETA=GSTARR/GR
      SQB=DSQRT(BETA)
      DPDA=-ASTARR*GSTARR
      DPDA=-GSTARR*(ASTARR+RAT*USTAR*CSTARR/(GR*   SQB))
      G41=1.D0/G4+0.5D0
      DRODA=(DRDXIR-DPDXIR/(CR*CR))   *BETA**G41+DPDA/(CSTARR**2)
      DRDXIX=   DRODA/SQB-DPIDT/(GSTARR*CSTARR**2)
      DRODA=DPDA/CSTARR**2-(RSTARR/(GAMA*SR))*DSDASR
      DRODTX=-DUDXIX*ROX**2
      DRDXIX=DRODA/SQB-DRODTX/GSTARR
 43   CONTINUE
      DUDTX=DUIDT
      DPDTX=DPIDT
      GO TO 9
 9    CONTINUE
C***********************************************************************
C  FLUXES CENTERED AT TIME T(N+1/2) AT EULERIAN POINT X=X(I).
C***********************************************************************
      FI1=ROX*UX
      FI2=ROX*UX**2+PX
      FI2=FI2-PX
      FI3=UX*(G12*PX+0.5D0*ROX*UX**2)
      FI4=ZX*ROX*UX
      FI3=FI3+QDET*FI4
      ROU00=ROX*UX
      GO TO(10,20,30,40,50,60), NFLUX
 10   CONTINUE
 60   CONTINUE
      DFDXI1=DRDXIX*UX+ROX*DUDXIX
      DFDXI2=DRDXIX*UX**2+2.D0*ROX*UX*DUDXIX+DPDXIX
      DFDXI2=DFDXI2-DPDXIX
      DFDXI3=DUDXIX*(G12*PX+0.5D0*ROX*UX**2)
     1      +UX*(G12*DPDXIX+0.5D0*DRDXIX*UX**2+ROX*UX*DUDXIX)
      DFDXI4=ZX*DFDXI1+ROX*UX*DZDXIX
      DFDXI3=DFDXI3+QDET*DFDXI4
      DFIDT1=DRODTX*UX+ROX*DUDTX
      DFIDT2=DRODTX*UX**2+2.D0*ROX*UX*DUDTX+DPDTX
      DFIDT2=DFIDT2-DPDTX
      DFIDT3=DUDTX*(G12*PX+0.5D0*ROX*UX**2)
     1      +UX*(G12*DPDTX+0.5D0*DRODTX*UX**2+ROX*UX*DUDTX)
      DFIDT4=ZX*DFIDT1+ROX*ZX*DZDTX
      DFIDT3=DFIDT3+QDET*DFIDT4
      FIDOT1=-ROU00*DFDXI1+DFIDT1
      FIDOT2=-ROU00*DFDXI2+DFIDT2
      FIDOT3=-ROU00*DFDXI3+DFIDT3
```

```
      FIDOT4=-ROU00*DFDXI4+DFIDT4
      UXDOT=-ROU00*DUDXIX+DUDTX
      PXDOT=-ROU00*DPDXIX+DPDTX
      ROXDOT=-ROU00*DRDXIX+DRODTX
      ZXDOT=-ROU00*DZDXIX+DZDTX
      FIH1=FI1+DT2*FIDOT1
      FIH2=FI2+DT2*FIDOT2
      GIH=PX+DT2*PXDOT
      FIH3=FI3+DT2*FIDOT3
      FIH4=FI4+DT2*FIDOT4
      UXN=UX+DT*UXDOT
      PXN=PX+DT*PXDOT
      ROXN=ROX+DT*ROXDOT
      ZXN=ZX+DT*ZXDOT
      IF(ZXN.LT.0.) ZXN=0.
      GO TO 90
20    CONTINUE
      EV0=GL*DSQRT(BETA0)
 201  CONTINUE
      DFIDA1=DRODAX*UX+ROX*DUDAX
      DFIDA2=DRODAX*UX**2+2.D0*ROX*UX*DUDAX+DPDAX
      DFIDA2=DFIDA2-DPDAX
      DFIDA3=DUDAX*(G12*PX+0.5D0*ROX*UX**2)
     1       +UX*(G12*DPDAX+0.5D0*DRODAX*UX**2+ROX*UX*DUDAX)
      DFIDA4=ZX*DFIDA1+ROX*UX*DZDAX
      FIDOT1=-EV0*DFIDA1
      FIDOT2=-EV0*DFIDA2
      FIDOT3=-EV0*DFIDA3
      FIDOT4=-EV0*DFIDA4
      FIH1=FI1+DT2*FIDOT1
      FIH2=FI2+DT2*FIDOT2
      FIH3=FI3+DT2*FIDOT3
      FIH4=FI4+DT2*FIDOT4
      GA=DGDAX
      IF(NFLUX.EQ.5)GA=-GA
      DROUA=UX*DRODAX+ROX*DUDAX
      BETAPR=0.5D0*DSQRT(BETA0)*(GA-DROUA)
      FIH2=FIH2-DPDBX*BETAPR*DT2
      UXDOT=-EV0*DUDAX+BETAPR*DUDBX
      PXDOT=-EV0*DPDAX+BETAPR*DPDBX
      GIH=PX+DT2*PXDOT
      ROXDOT=-EV0*DRODAX+BETAPR*DRODBX
      ZXDOT=-EV0*DZDAX
      UXN=UX+DT*UXDOT
      PXN=PX+DT*PXDOT
      ROXN=ROX+DT*ROXDOT
      ZXN=ZX+DT*ROXDOT
      IF(ZXN.LT.0.) ZXN=0.
      GO TO 90
50    CONTINUE
      EV0=-GR*DSQRT(BETA0)
      GO TO 201
30    CONTINUE
40    CONTINUE
      GO TO 60
90    CONTINUE
      RETURN
      END
```

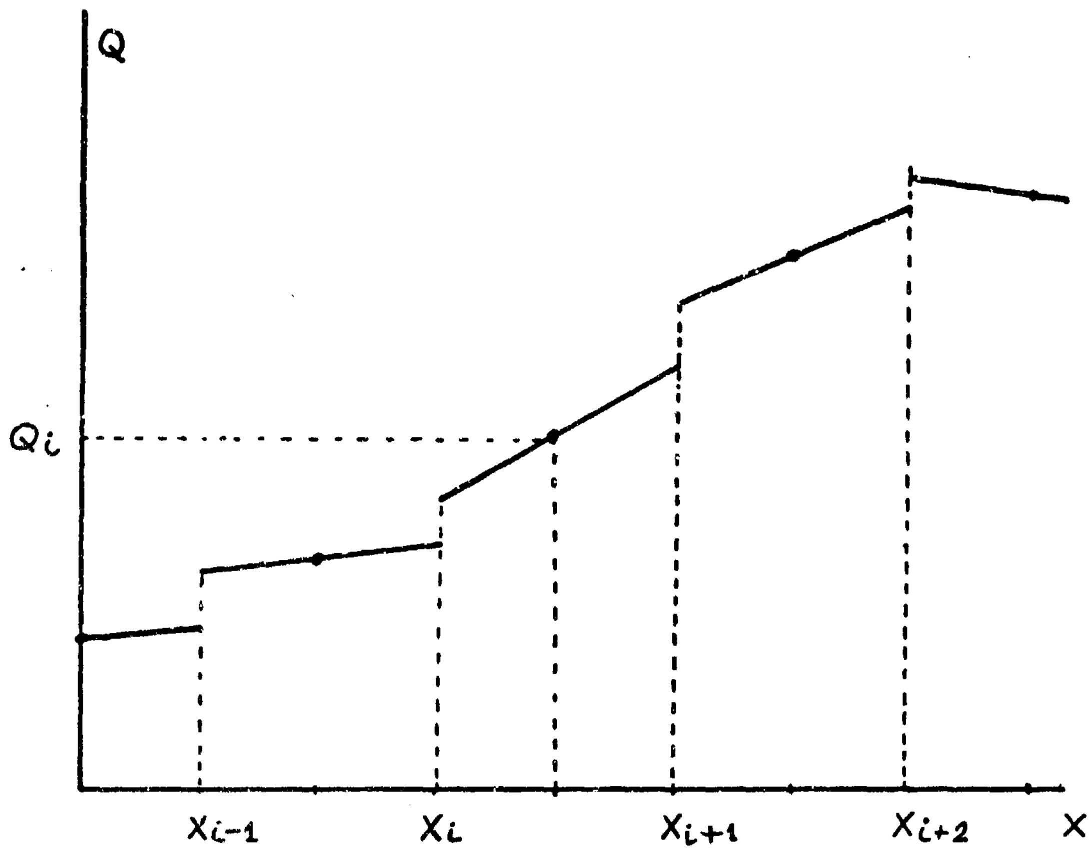

Figure A-1. Piecewise Linear Distribution of Flow Variables in Cells

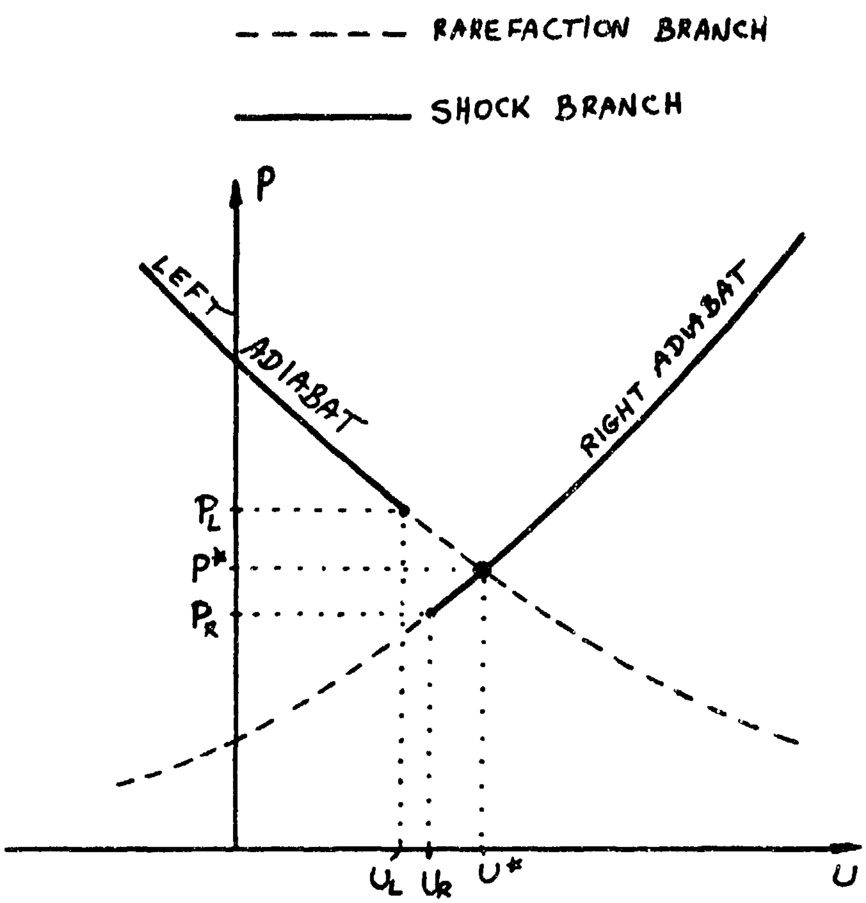

Figure A-2. Intersection of Right and Left Adiabats for Solving Riemann Problem

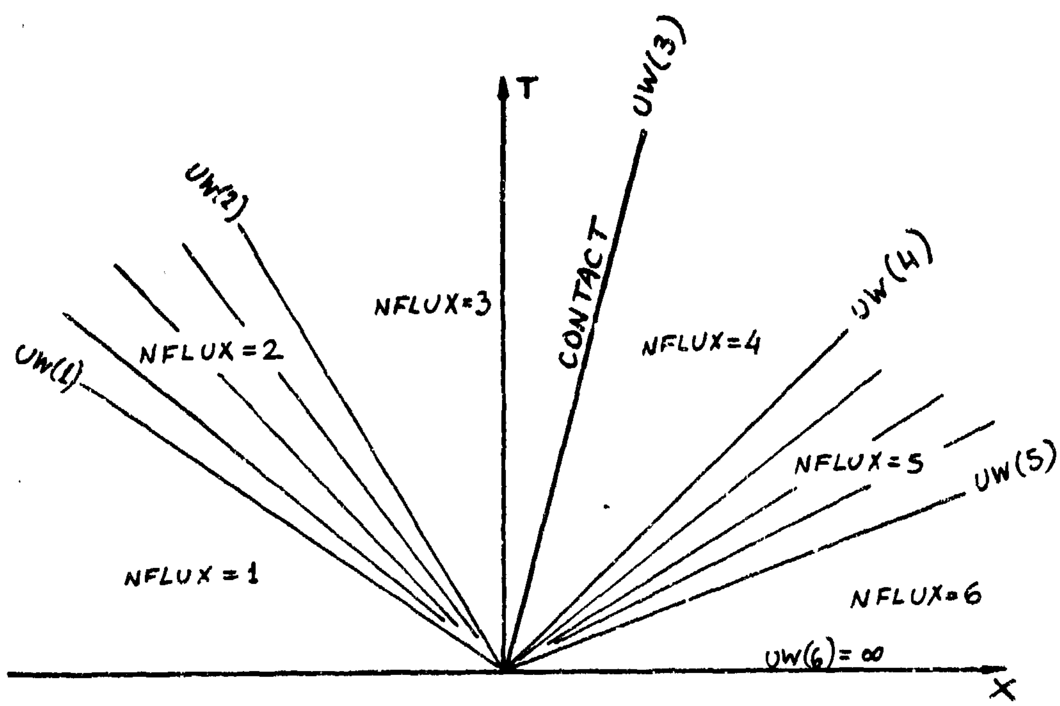

Figure A-3. Wave Diagram Representing Solution to Riemann Problem

APPENDIX B. Code for Re-Normalizing the Air Impulse

```
      IMPLICIT REAL*8(A-H,O-Z)                                          REN00010
C  CODE RENORM -- C  TRANSFORMATION OF TOTAL REFLECTED IMPULSE FROM     REN00020
C  BAKER'S CHART TO SPACE-NORMALIZED VALUES.                            REN00030
C  DATA FROM FIG. 6.3 (SUPPLEMENT) IN BAKER'S BOOK "EXPLOSIONS IN AIR"  REN00040
      REAL*4 RB,IB,RS,IS,ISBARE                                         REN00050
      DIMENSION RB(21),IB(21)                                           REN00060
      DIMENSION RS(21),IS(21),ISBARE(21)                                REN00070
      DATA RB/.05,.06,.07,.08,.09,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.,        REN00080
     1        2.,3.,4.,5.,6.,7./                                        REN00090
      DATA IB/4.4,3.06,2.30,1.83,1.50,1.27,.457,.293,.221,.178,.149,    REN00100
     1        .128,.113,.099,.0885,.0376,.0236,.0173,.0136,.0113,.0095/ REN00110
      PAI=4.D0*DATAN(1.D0)                                              REN00120
      G=1.4D0                                                           REN00130
      PA=0.1D0                                                          REN00140
      RHOA=1.3D0                                                        REN00150
      RHO0=1800.D0                                                      REN00160
      Q0=4.D0                                                           REN00170
      BETA=DSQRT(RHOA/RHO0)*(PA/(RHO0*Q0))**(1.D0/6.D0)                 REN00180
      GOREM=(3.D0/DSQRT(2.D0*G))*(4.D0*PAI/3.D0)**(1.D0/3.D0)           REN00190
      BETA=BETA*GOREM                                                   REN00200
      DELTA=( (4.D0*PAI/3.D0)*(RHO0*Q0/PA) )**(1.D0/3.D0)               REN00210
      PRINT 11, BETA,DELTA                                              REN00220
   11 FORMAT(/1X,'RESULTS WITH  BETA,DELTA=',2D16.7//                   REN00230
     1        1X,'   N',' RB       ',' IB        ',2X,                  REN00240
     2           '   RS      ','  IS      ',2X,'   ISBARE   '/)         REN00250
                                                                        REN00260
      DO 1 N=1,21                                                       REN00270
      RS(N)=RB(N)*DELTA                                                 REN00280
      IS(N)=IB(N)*BETA                                                  REN00290
      ISBARE(N)=1.D0/RS(N)**2                                           REN00300
      PRINT 2, N,RB(N),IB(N),RS(N),IS(N),ISBARE(N)                      REN00310
    2 FORMAT(1X,I4,2E12.4,2X,2E12.4,2X,E12.4)                           REN00320
    1 CONTINUE                                                          REN00330
      END                                                               REN00340
```

RESULTS WITH BETA,DELTA= 0.1204163D-01 0.6706157D+02

N	RB	IB	RS	IS	ISBARE
1	0.5000E-01	0.4400E+01	0.3353E+01	0.5298E-01	0.8894E-01
2	0.6000E-01	0.3060E+01	0.4024E+01	0.3685E-01	0.6177E-01
3	0.7000E-01	0.2300E+01	0.4694E+01	0.2770E-01	0.4538E-01
4	0.8000E-01	0.1830E+01	0.5365E+01	0.2204E-01	0.3474E-01
5	0.9000E-01	0.1500E+01	0.6036E+01	0.1806E-01	0.2745E-01
6	0.1000E+00	0.1270E+01	0.6706E+01	0.1529E-01	0.2224E-01
7	0.2000E+00	0.4570E+00	0.1341E+02	0.5503E-02	0.5559E-02
8	0.3000E+00	0.2930E+00	0.2012E+02	0.3528E-02	0.2471E-02
9	0.4000E+00	0.2210E+00	0.2682E+02	0.2661E-02	0.1390E-02
10	0.5000E+00	0.1780E+00	0.3353E+02	0.2143E-02	0.8894E-03
11	0.6000E+00	0.1490E+00	0.4024E+02	0.1794E-02	0.6177E-03
12	0.7000E+00	0.1280E+00	0.4694E+02	0.1541E-02	0.4538E-03
13	0.8000E+00	0.1130E+00	0.5365E+02	0.1361E-02	0.3474E-03
14	0.9000E+00	0.9900E-01	0.6036E+02	0.1192E-02	0.2745E-03
15	0.1000E+01	0.8850E-01	0.6706E+02	0.1066E-02	0.2224E-03
16	0.2000E+01	0.3760E-01	0.1341E+03	0.4528E-03	0.5559E-04
17	0.3000E+01	0.2360E-01	0.2012E+03	0.2842E-03	0.2471E-04
18	0.4000E+01	0.1730E-01	0.2682E+03	0.2083E-03	0.1390E-04
19	0.5000E+01	0.1360E-01	0.3353E+03	0.1638E-03	0.8894E-05
20	0.6000E+01	0.1130E-01	0.4024E+03	0.1361E-03	0.6177E-05
21	0.7000E+01	0.9500E-02	0.4694E+03	0.1144E-03	0.4538E-05

6. DISTRIBUTION LIST

		No. of Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5100	2
3.	Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, CA 93943-5100	1
4.	Distinguished Professor Allen E. Fuhs Space Systems Academic Group, Code 72 Naval Postgraduate School Monterey, CA 93943-5100	5
5.	Dr. Neil Griff SDIO/DEO Washington, DC 20301-7100	3
6.	Mr. Bruce Pierce SDIO/DEO Washington, DC 20301-7100	1
7.	Dr. Joseph Falcovitz Code 72 Naval Postgraduate School Monterey, CA 93943-5100	5

No. of Copies

8. Professor R. E. Ball
Department of Aeronautics, Code 67
Naval Postgraduate School
Monterey, CA 93943-5100. 1

9. Professor Y. S. Shin
Department of Mechanical Engineering, Code 69
Naval Postgraduate School
Monterey, CA 93943-5100. 1

10. Research Administration Office
Code 012
Naval Postgraduate School
Monterey, CA 93943-5100. 1

11. Dr. P. Avizonis
Air Force Weapons Laboratory
Kirtland Air Force Base, NM 87117 1

12. Dr. John Lawless
Space Power Inc.
1977 Concourse Drive
San Jose, CA 95131 1

13. Dr. Mark Thornton
Boeing Aerospace Company
Post Office Box 3999
Seattle, WA 98124-2499 1

14. LT. Mark Price
AFRPL
Edwards AFB, CA 93523 1

No. of Copies

15. Mr. Arthur W. Rogers
    Space Systems Division
    Hughes Aircraft Co.
    P. O. Box 92919, Los Angeles, CA 90009 1

16. LCOL Rick Babcock, USAF
    Air Force Geophysical Laboratory
    Hanscomb Field
    Bedford, MA 01730 1